

UNCLASSIFIED



***Integrated Sensing and Processing (ISP) Phase II:
Demonstration and Evaluation for Distributed Sensor
Networks and Missile Seeker Systems***

FINAL REPORT

29 May 2007

Acknowledgment of Support

This material is based upon work supported by the United States Air Force under Contract No. N00014-04-C-0437.

Contract No.: N00014-04-C-0437
Contract Line Item Number 0001
Deliverable Item: Publications (A004-001)

Raytheon Company
P.O. Box 11337
Tucson, AZ 85734-1337

DESTRUCTION NOTICE

Distribution Statement A.

Approved for public release.
Distribution is unlimited.

For classified documents, follow the procedures in DOD 5220.22M, National Industrial Security Program Operating Manual (NISPOM), Chapter 5, Section 7, or DOD 5200.1-R, Information Security Program Regulation, Chapter IX. For unclassified, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.

© RAYTHEON MISSILE SYSTEMS (2007) UNPUBLISHED WORK

This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227 7013 (Nov 1995)

UNCLASSIFIED

Report Documentation Page			<i>Form Approved OMB No. 0704-0188</i>	
<p>Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p>				
1. REPORT DATE 29 MAY 2007	2. REPORT TYPE	3. DATES COVERED 00-00-2007 to 00-00-2007		
4. TITLE AND SUBTITLE Integrated Sensing and Processing (ISP) Phase II: Demonstration and Evaluation for Distributed Sensor Networks and Missile Seeker Systems				
5a. CONTRACT NUMBER				
5b. GRANT NUMBER				
5c. PROGRAM ELEMENT NUMBER				
6. AUTHOR(S)				
5d. PROJECT NUMBER				
5e. TASK NUMBER				
5f. WORK UNIT NUMBER				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Raytheon Company, PO Box 11337, Tucson, AZ, 85734-1337				
8. PERFORMING ORGANIZATION REPORT NUMBER				
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				
10. SPONSOR/MONITOR'S ACRONYM(S)				
11. SPONSOR/MONITOR'S REPORT NUMBER(S)				
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited				
13. SUPPLEMENTARY NOTES The original document contains color images.				
14. ABSTRACT				
15. SUBJECT TERMS				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 68
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified	19a. NAME OF RESPONSIBLE PERSON	



Raytheon Systems Company

29 May 2007

Progress Report

CDRL A004 No. 1

Final Technical and Progress Report for Period of Performance
24 Feb 2005 – 31 May 2007

Integrated Sensing Processor Phase 2

Program Manager: Dr. Harry A. Schmitt

Principal Investigator: Dr. Harry A. Schmitt

Sponsored By:

Defense Advanced Research Projects Agency/DSO

Dr. Carey Schwartz/DARPA DSO

Program Manager: Dr. Dan Purdy/ONR

Issued by ONR under Contract #N00014-04-C-0437

Prepared By:

Raytheon Systems Company

P.O. Box 11337

Tucson, AZ 85734

EXECUTIVE SUMMARY

The primary goal of this effort is to bring to maturity a select set of basic algorithms, hardware, and approaches developed under the Integrated Sensing and Processing (ISP) Phase I program, implement them on representative hardware, and demonstrate their performance in a realistic field environment. We have identified a few promising research thrusts investigated in ISP Phase I where field demonstrations are cost prohibitive but collected data sets are available. Here, we will conduct a thorough performance evaluation.

TABLE OF CONTENTS

0. Technical Abstract	5
1.0. Management Overview and Summary.....	5
1. A. Program Summary	5
1. B. Program Status	6
1. C. Personnel Associated/Supported	6
1. D. Recent Events.....	7
1. E. Near Term Events.....	7
2. 0. Technical Progress.....	7
2.A. Preliminaries and Background	7
2.A.1. Use of CCCDs for Informative Feature-Space Partitioning	9
2.A.2. Information Theoretic Approaches for Radar Threat Identification	18
2.A.3 ASU Technical Progress	25
2.A.4. Distributed Mote Tracker Support	33
2.A.5. Scheduling of Multiple UAV Platforms for Passive Geolocation	42
2.A.6 UniMelb Technical Progress.....	51
2.A.7 University of Michigan Technical Progress.....	59
2.A.8 Georgia Tech Technical Progress	59
2. B. Refereed Publications.....	60
2. C. Conference Proceedings.....	60
2. D. Consultative and Advisor Functions	60
2. E. New Discoveries, Inventions or Patent Disclosures.....	61
2. F. Honors/Awards	61
2. G. Transitions.....	61
2. I. Acronyms.....	61
2. J References	62

INDEX OF FIGURES

Figure 1: ISP II Quad Chart	5
Figure 2: Spin Image Geometry.....	11
Figure 3: Spin Images at Various Vehicle Surface Locations	12
Figure 4: Estimates over 10 runs of HPD as a function of sample size for dissimilar targets, $h(T_1, T_2)$ and for similar targets, $h(T_2^*, T_2)$	14
Figure 5: Histograms and scatterplots of population and radii of balls in class covers for dissimilar targets (top row) and similar targets (bottom row), with $n = 2000$. CCCD parameter values: $\alpha = \beta = 0$	15
Figure 6: Histograms and scatterplots of population and radii of balls in class covers for dissimilar targets (top row) and similar targets (bottom row), with $n = 2000$. CCCD parameter values: $\alpha = 0, \beta = 0.01$	16
Figure 7: Histograms and scatterplots of population and radii of balls in class covers for dissimilar targets (top row) and similar targets (bottom row), with $n = 2000$. CCCD parameter values: $\alpha = 0, \beta = 0.05$	16
Figure 8: For different targets, estimated divergence as a function of α	17
Figure 9: For similar targets, estimated divergence as a function of α	17
Figure 10: Singular values on the left, and differences between computed and hand solution on the right, for a simple wave.....	22

Figure 11: Singular values on the left, and differences between computed and hand solution on the right, for a simple wave.....	23
Figure 12: Comparison of two simple signals	24
Figure 13: comparison of a simple signal and a composite signal.	24
Figure 14: Footstep energy as function of distance: means & confidence intervals	26
Figure 15: P_D and P_{FA} in the presence of interference	27
Figure 16: Sensor grid with target trajectory; the grid spacing is in meters.	27
Figure 17: 1-Bit Tracker Performance	28
Figure 18: Energy Tracker Performance.....	28
Figure 19: Footstep data at 2 feet from sensor.....	29
Figure 20: Zoomed version of a single footstep	29
Figure 21: Probability of a detection as a function of sensor/target distance.	30
Figure 22: Leader node scheduling problem scenario	31
Figure 23: Running average energy (RAE) as a function of time	32
Figure 24: MSE plot for the central node scheduling problem.....	32
Figure 25: Sensor cost for different scenarios in the central node scheduling problem...	33
Figure 26: Raytheon Distributed Tracking Test Site	33
Figure 27: Raytheon ISP Test Bed.....	34
Figure 28: Gateway Overview	35
Figure 29: ISP Phase II Test Bed MATLAB GUI.....	35
Figure 30: Sensor Localization using DWMDS	37
Figure 31: Particle Filter tracker tracking a beeping target	38
Figure 32: Particle Filter tracker tracking a footstep target.	39
Figure 33: Virtual Measurement tracker tracking a beeping target	40
Figure 34: Virtual Measurement tracker tracking a footstep target.....	40
Figure 35: UKF tracker tracking a beeping target	41
Figure 36: UKF tracker tracking a footstep target.....	42
Figure 37: Target location ambiguities arise when using only three sensors (the black squares labeled by: F1, F2 and F3). Three colored hyperbolic envelopes are shown, corresponding to the three pairs of focal points. Two solution regions are shown; these intersection regions are located near the points: (x~160,y~120) and (x~280,y~80).....	44
Figure 38: Four sensors (S1 through S4) plus a target (T) in a square geometry	44
Figure 39: Green lines indicate three TDOA hyperbolic constraint pairs	45
Figure 40: Red and blue lines (three each) indicate a total of six TDOA hyperbolic constraint pairs.	45
Figure 41: Plot of target position covariance ellipses for the basic scenario, obtained in two ways. The blue ellipses are computed from <i>all</i> 6 hyperbolic TDOA constraints, while the red ellipses are computed from <i>only</i> 3 such constraints (<i>i.e.</i> , TDOAs between sensors located at 0,0 and 1,0; 0,0 and 0,1; and 0,0 and 1,1)	50
Figure 42: Emitter position uncertainty. Two UAVs, four pulse uncertainty	52
Figure 43: Emitter position uncertainty. Three UAVs, four pulse uncertainty	52
Figure 44: One hyperbola, measurement uncertainty presentation	53
Figure 45: Highest probability track component uncertainty after 5 TDOA measurements. Units are in km.....	54

ISP Phase II (Contract N00014-04-C-0437)
Final Technical Progress Report (CDRL A004 No. 1)

Figure 46: Highest probability track component uncertainty after 6 TDOA measurements. Units are in km.....	55
Figure 47: RMS estimation errors over number of TDOA measurements. Ordinate units are in km	55
Figure 48: Detection and tracking using the UKF with real data: (a) data file 20070422205553_10m_test1, (b) data file 20070422205553_10m_test2. The red solid line is the true target trajectory. The green dotted line is the estimated trajectory of any confirmed targets.....	56
Figure 49: Tracking results with a restricted number of sensors. The criterion of equation (1) is used with $\Lambda = k P_{fa}$ where (a) $k=2^5$, (b) $k=2^{10}$, (c) $k=2^{15}$ and (d) $k=2^{20}$	57
<u>INDEX OF TABLES</u>	
Table 1: Balls in class covers, mean and standard deviation over 10 runs, for n=2000. CCCD parameter values: $\alpha = \beta = 0$	14
Table 2: Balls in class covers, mean and standard deviation over 10 runs, for n=2000. CCCD parameter values: $\alpha = 0$; $\beta = 0.01$	15
Table 3: Balls in class covers, mean and standard deviation over 10 runs, for n=2000. CCCD parameter values: $\alpha = 0$; $\beta = 0.05$	15

0. Technical Abstract

Advances in sensor technologies, computation devices, and algorithms have created enormous opportunities for significant performance improvements on the modern battlefield. Unfortunately, as information requirements grow, conventional network processing techniques require ever-increasing bandwidth between sensors and processors, as well as potentially exponentially complex methods for extracting information from the data. To raise the quality of data and classification results, minimize computation, power consumption, and cost, future systems will require that the sensing and computation be jointly engineered. Integrated Sensing and Processing (ISP) is a philosophy/methodology that eliminates the traditional separation between physical and algorithmic design. By leveraging our experience with numerous sensing modalities, processing techniques, and data reduction networks, the Raytheon team will develop ISP into an extensible and widely applicable paradigm. The improvements we intend to demonstrate here are applicable in a general sense; however, this program will focus on distributed sensor networks and missile seeker systems. The Raytheon Company, Missile Systems (Raytheon) ISP II Demonstration and Evaluation program is summarized in the Quad Chart (c.f., Figure 1).

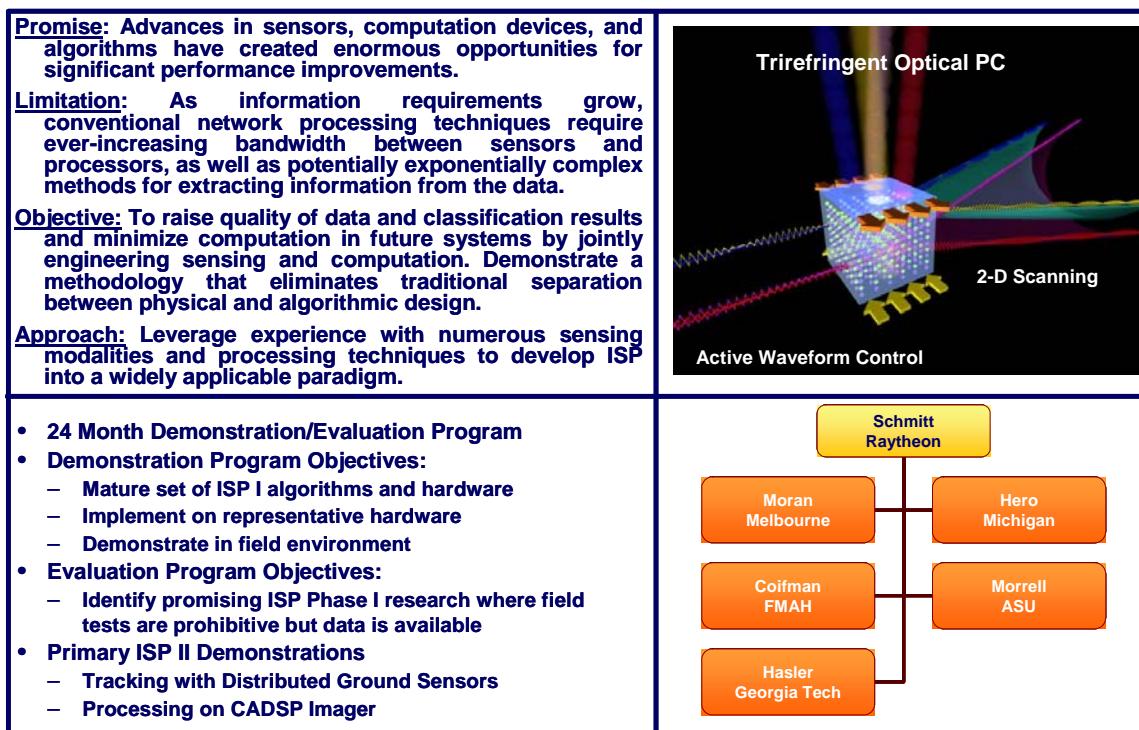


Figure 1: ISP II Quad Chart

1.0. Management Overview and Summary

1. A. Program Summary

The Raytheon Company, Missile Systems (Raytheon) ISP Phase II program is a twenty-four month contract with an original Period of Performance (PoP) covering 1 March 2005 to 28 February 2007. Raytheon has asked for and received a two month no-cost extension, resulting in final completion date of 31 May 2007. Raytheon has four

ISP Phase II (Contract N00014-04-C-0437)
Final Technical Progress Report (CDRL A004 No. 1)

universities and one small business as ISP Phase II subcontractors: Arizona State University (ASU); Fast Mathematical Algorithms and Hardware (FMAH); Georgia Institute of Technology (Georgia Tech); Melbourne University (UniMelb) and the University of Michigan (UM).

1. B. Program Status

The Raytheon ISP Phase II Program status can be summarized as remaining “on track” for a completion date of 31 May 2007. All subcontractors are expected to finish on budget. Two subcontractors – Georgia Tech and UniMelb – are currently significantly under-spent. In both cases, the under-runs are due to a time lag in billing the contract. We expect to complete the Prime contract on budget. We had incurred some minor schedule slips on both the distributed tracking and the Georgia Tech Cooperative Analog Digital Signal Processing (CADSP) imager demonstrations during the latest PoP. These schedule slip resulted in the final demonstrations being held at Raytheon on 23 April 2007. This is a slip of approximately two months since the last Technical Report.

1. C. Personnel Associated/Supported

The personnel supporting the Raytheon ISP Phase II have remained stable over the last several PoPs. Key personnel and their associated organizations are summarized below.

Raytheon

Dr. Harry A. Schmitt	Principal Investigator
Mr. Donald E. Waagen	Co-Principal Investigator
Dr. Sal Bellofiore	Distributed Sensing Lead
Mr. Thomas Stevens	Distributed Sensing Support
Dr. Robert Cramer	Mathematical Support
Mr. Craig Savage	Waveform Design and Control Lead
Dr. Nitesh Shah	High Dimensional Processing Data Lead

FMAH

Dr. Paolo Barbano
Professor Ronald Coifman
Dr. Nicholas Coul

ASU

Professor Darryl Morrell
Professor Antonia Papandreou-Suppappola

Georgia Tech

Professor David Anderson
Professor Paul Hasler

UniMelb

Dr. Barbara LaScala
Professor William Moran
Dr. Darko Musicki
Dr. Sofia Suvorova

UM

Professor Al Hero
Dr. Raviv Raich

Significant Personnel Actions: There were no significant personnel changes during the current PoP.

1. D. Recent Events

The following events occurred during the current PoP:

- The final Raytheon ISP Phase II Program Review and Demonstrations was held at Raytheon Missile Systems on 23 April 2007. In attendance were Sal Bellofiore, Bob Cramer, Harry Schmitt, Nitesh Shah, Thom Stevens and Don Waagen (Raytheon); Darryl Morrell (ASU); Bill Moran (UniMelb); David Anderson (Georgia Tech) and Jeff Farrell (Booz Allan Hamilton). Also attending were Capt. Matt Hyland, Eleanor Gillis and Renee Bousselaire (DARPA Intern Program)
- Harry Schmitt, Don Waagen (Raytheon) and Bill Moran (Melbourne) visited DARPA MTO (Dennis Healy) and DARPA SPO (Ed Baranowski) to review Raytheon ISP Phase II accomplishments and discuss future research directions.
- Don Waagen (Raytheon) met with Dr. T. J. Klausutis (AFLR/Eglin) to discuss recent research in manifold alignment techniques as applied to Synthetic Aperture Radar and LADAR.
- Don Waagen (Raytheon), Paul Hasler and David Anderson (Georgia Tech), Al Hero (Michigan), and Antonia Papandreou-Suppappola (ASU) attended the 2007 International Conference Acoustics Speech and Signal Processing (ICASSP 2007) conference 15 – 20 April 2007 in Honolulu, Hawaii.

1. E. Near Term Events

- The Raytheon ISP Phase II Final Technical Report is due on 31 May 2007.
- One of the Georgia Tech CADSP Imagers will be delivered to Raytheon. The final delivery date is still to be determined.
- Harry Schmitt (Raytheon) will visit UniMelb in May 2007.

2. 0. Technical Progress

In this section we provide a more detailed discussion of the technical progress that occurred during the current PoP broken down by subcontractor. We first provide a brief review for context.

2.A. Preliminaries and Background

As mentioned previously, the Raytheon ISP Phase II program consisted of several demonstrations and evaluations. The two primary demonstrations were personnel tracking using a distributed network of wireless sensors (MICA-2 Motes [1]) and image processing using the Georgia Tech CADSP imager.

The distributed tracking demonstration required theoretical advances in three primary areas: (i) mote self-localization; (ii) on-mote detection; and (iii) 1-bit base station trackers.

1. Mote self-localization is a key research area and critical to the eventual deployment of large scale ad hoc networks (especially in GPS-denied environments). We have investigated two approaches: (i) the UM algorithm based on Received Signal Strength (RSS); and (ii) the UniMelb algorithm that is based on the Vanderbilt University (VU) Radio Interferometric Positioning System (RIPS [2]) algorithm. While mote self-localization was not considered part of the official demonstration, we evaluated the performance of our ISP Phase algorithms as follows. The VU RIPS was taken as state-of-the-art in accuracy (~5 cm), but is not implementable. RIPS scales exponentially in mote number, its Genetic Algorithm estimation approach converges poorly and it can produce multiple solutions. While a performance trade space is complex, our performance goal was scalability with accuracy comparable to the VU Acoustic Ranging algorithm (~1 m). The MU RSS algorithm was integrated with our mote test bed (*c.f.*, Section 2.A.1); field performance was ~3× worse than the ~50 cm reported by MU in less representative test environment. In addition, scalability of the current RSS algorithm is limited by on-mote memory constraints. The MU RSS algorithm has been discussed in depth in previous Technical Reports and will not be covered here. The UniMelb RIPS-Based algorithm has not been integrated with our mote test bed, but its preliminary performance (both scalability and accuracy) are extremely promising based on and simulated and collected RIPS data. The UniMelb algorithm is discussed in depth in Section 2.A.3.
2. ASU was responsible for developing the on-mote detection algorithms. ASU has developed two implementations – a 1-bit threshold detector and an 8-bit energy detector. The 1-bit detector has been integrated with our mote test bed, while the 8-bit detector has been evaluated outside of the test bed. Here, we have taken the baseline detector as one that transmits all 8-bit to a tracker. The tracker employed here was an 8-bit extension to the ASU 1-bit particle filter tracker which has been described in detail in earlier reports. Our objective was to demonstrate an increase in Tracker Mean Squared Error (MSE) of less than 25% using only the 1-bit detector (*i.e.*, bandwidth constrained tracking performance). The ASU work is discussed in more detail in Section 2.A.2.
3. There were three 1-bit trackers developed over the course of the Raytheon ISP Phase II program. These include the Virtual Measurement (VM) and Unscented Kalman Filter (UKF) trackers developed by UniMelb and the Particle Filter tracker developed by ASU. All three trackers have been discussed extensively in previous Technical Reports. All three trackers have been integrated with our mote test bed and were run during the 23 April final demonstration. Our objective was to demonstrate an increase in Tracker MSE of less than 10% with only 50% of the mote sensors continuously active (*i.e.*, power constrained tracking performance). Details of the trackers performance are provided in Section 2.A.1, 2.A.2 and 2.A.3 below.

The Georgia Tech CADSP image processing demonstration is intended to illustrate the mixed Analog-Digital nature of the imager – specifically, the CADSP imager was selected for inclusion in the DARPA ISP Phase II program because it enables analog processing under digital control. The CADSP imager was designed to be able to

perform separable 2-D linear operations of form $A^T PB$ and to have access to multiple (4) stored A and B matrices. The A and B matrices are to be of sufficient size to implement various transforms. The CADSP imager demonstration consisted of the following components:

1. Perform JPEG compression and associated preprocessing an image. At one level this demonstration is to test the functionality of the CADSP imager. The image preprocessing step will verify that the pseudo non-uniformity compensation (NUC) that Georgia Tech implements is sufficient. The correctness of the CADSP JPEG compression implantation will be demonstrated by comparison to a standard DSP approach. The more interesting ISP application consists of a power constrained JPEG Compression. Here, our objective is to achieve equal compression performance with $\geq 90\%$ reduction in power consumption.
2. An application that is well suited for implementation on the CADSP imager is optical flow processing. Optical flow is one popular tracking approach being considered for Unmanned Aerial Vehicles; in this case the low power consumption of the CADSP imager is attractive. We have investigated a bandwidth constrained foveal tracker implementable on the CADSP imager. The objective is to show equivalent Tracker MSE with 80% reduction in bandwidth.
3. To insure the utility of the CADSP imager an API will be developed that includes testing steps required for imager validation.

In addition to the above mentioned demonstrations, the Raytheon ISP Phase II program also had the objective of evaluating some of the promising signal processing approaches developed under ISP Phase I. The algorithms included, but were not limited to, sensor scheduling, sensor allocation and manifold extraction/dimensionality reduction. Raytheon has been particularly successful in applying ISP-developed high dimensionality data processing algorithms to a wide variety of collected data sets that were collected with a number of different sensor modalities. We have reported in detail on a number of these activities in previous Quarterly Technical Reports. Here, we report in depth on three specific research topics considered by Raytheon personnel: (i) scheduling of multiple UAVs for passive geolocation; (ii) Class Cover Catch Diagraphs (CCCD) for feature-space partitioning; and (iii) information theoretic approaches to radar threat identification (TID).

2.A.1. Use of CCCDs for Informative Feature-Space Partitioning

2.A.1.1. Background

2.A.1.1.1. Introduction

In high-dimensional settings, direct estimation of class-conditioned probability densities is generally constrained in practice by limited class sample sizes and the associated curse of dimensionality. However, estimating the similarities and differences of feature distributions is necessary for evaluating feature utility for potential classification efficacy. Therefore an efficient and effective approach for partitioning high-dimensional spaces into “informative” regions is of interest. In this regard, we investigate application of the class cover catch digraph (CCCD) algorithm [3], [4]. An evaluation of the potential classification efficacy of the partitions produced is

demonstrated using spin image representations of point clouds sampled from Computer-Aided-Design (CAD) models.

Our dataset consists of three-dimensional (3D) surface coordinates derived from CAD models of three military targets. This dataset serves as a surrogate for field-measured 3D Laser Detection and Ranging (LADAR) point-cloud data. The surface coordinates are processed into spin-image stacks [5], [6]. In both the training and testing spaces, we measure partition efficacy and statistical characteristics using distributional separability (divergence). For measuring distributional separation, we use the Henze-Penrose Divergence (HPD) [7], [8].

For many datasets, features are not global concepts measured from an object, rather they are extracted from localized regions of the object of interest. For instance, spin images form a collection of feature vectors, each vector representing the local geometric structure of a part of the object of interest. Often, much of the structural information and content is shared between objects of interest. For example, most cars have four wheels. Extraction of features corresponding to wheels of equal size will be noninformative in aiding a user in classification of a wheeled vehicle. (Note that wheels might be informative for detection of the vehicle, and in recognizing the vehicle as wheeled rather than tracked, but those are *different* questions). Our application of the CCCD approach pertains to the partitioning of a collection of feature vectors into partitions that are more informative or less informative, with respect to recognition / classification.

Classifier complexity can be driven by several factors, including sparse sampling (curse of dimensionality), complex high-dimensional data structure / decision boundary, and regions of ambiguity (due either to the intrinsic nature of the problem or to poor selection of features) [9]. When there are more than approximately 5 dimensions/labels/features, the dataset takes on a “high-dimensional” nature [10]. Limited sample support in high-dimensional spaces leads to the well known curse of dimensionality. However one is generally given a fixed-size dataset, and it is not usually feasible to increase the sampling density. Given the dataset, we seek to characterize the representation complexity, due to both high-dimensional decision boundaries and regions of ambiguity. The decision boundary complexity is measured using the HPD, and as previously discussed the regions of class distinctiveness and class ambiguity are explored using the CCCD algorithm.

Distribution-based comparisons of feature-set efficacy are prone to error, whether the distribution estimation is based on density-building using kernels or fitting parameters in a predefined model. There are some methods available for distribution-free, approximate measurement of feature-set divergence (or separability). In the two-class case, the measurement of feature-set efficacy can be recast in terms of the general multivariate two-sample problem. The underlying assumption is that independent of any classifier, feature sets that exhibit more divergence (or separability) should in general be of greater utility for classification than feature sets that exhibit less divergence (or separability). In [11] and [9], the Friedman-Rafsky statistic is suggested as a boundary-description-based measure of separability of classes, and in [9] adhesion classes [12] are suggested as an interior description for the local clustering properties of the various

classes. We note that the adhesion-class pretopology approach is a precursor to the more general CCCD approach.

2.A.1.1.2. Spin images

Spin images provide a pose-invariant statistical encoding of the surface shape of objects for which measurements are available as a uniformly sampled surface mesh. A spin image is a two-dimensional (2D) histogram computed at an oriented vertex \mathbf{P} located on the surface mesh of an object. The histogram accumulates the coordinates μ and v (defined below) of all other surface mesh vertices \mathbf{Q} that are located within the support region. With $\hat{\mathbf{S}}_{\mathbf{P}}$ representing the surface normal at \mathbf{P} and $\mathbf{T}_{\mathbf{P}}$ representing the tangent plane at \mathbf{P} , the support region is defined as follows (see **Error! Reference source not found.**):

- The coordinate μ , the distance from \mathbf{P} to the projection of \mathbf{Q} onto $\mathbf{T}_{\mathbf{P}}$, obeys the inequality $0 \leq \mu \leq \mu_{\max}$, where μ_{\max} is the user-defined *support cylinder radius*.
- The coordinate v , the distance from \mathbf{Q} to $\mathbf{T}_{\mathbf{P}}$, satisfies $-0.5v_{\max} \leq v \leq 0.5v_{\max}$, where v_{\max} is the user-defined *support cylinder height*.
- The angle φ between $\hat{\mathbf{S}}_{\mathbf{P}}$ and $\hat{\mathbf{S}}_{\mathbf{Q}}$ (the surface normal at \mathbf{Q}) obeys the inequality $\varphi \leq \varphi_{\max}$, where φ_{\max} is the user-defined *support angle*.

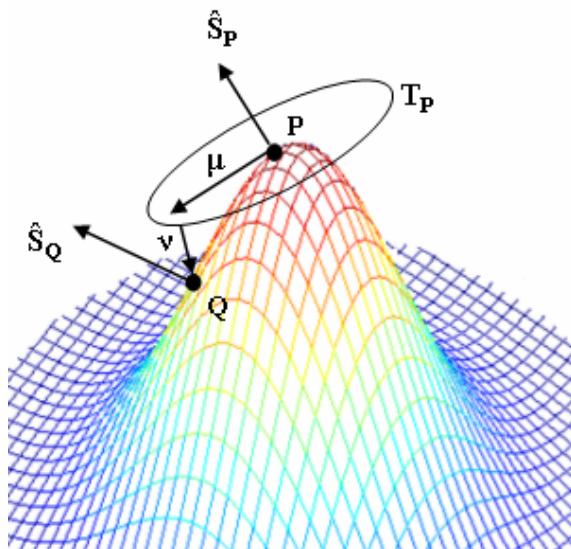


Figure 2: Spin Image Geometry

Spin images formed from small support regions (local information only) provide more robustness to clutter and occlusion, whereas spin images formed from large support regions (global information included) provide greater discrimination capability. The spin image's 2D histogram (see **Error! Reference source not found.**) can be raster-scanned into a spin-image vector with dimensionality equal to the number of bins in the histogram. For an object, the set of spin-image vectors forms a spin-image stack.

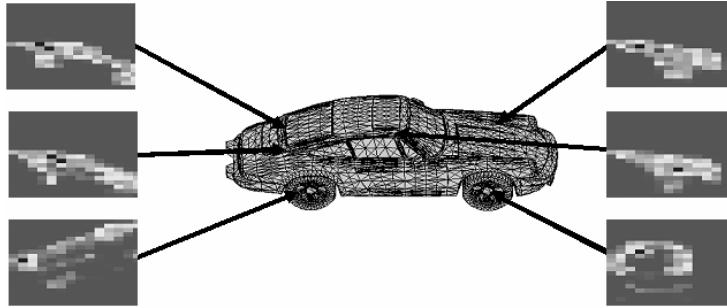


Figure 3: Spin Images at Various Vehicle Surface Locations

2.A.1.1.3. Class cover catch digraphs

In the CCCD approach, class-conditional regions are modeled with a mixture of d -dimensional balls. The number, location and size of the balls are determined based on the proximity between training samples. The balls potentially form a low-complexity representation of each class. Each class-cover ball contains a fixed number β of out-of-class samples (purity factor / sensitivity to contamination) and the union of class-cover balls can neglect up to α in-class samples (properness factor / sensitivity to outliers). Setting $\alpha = \beta = 0$ results in a proper, pure cover. In this work, we specify the parameters α and β as percentages of class-conditioned samples rather than as fixed integer values. Although the CCCD approach is extensible to an n -class case, for purposes of clarity of presentation we work here exclusively with two-class cases (either two similar objects, or two dissimilar objects).

By careful selection of α , one can limit the cover produced to regions of feature space where the classes are “well separated” as measured by distributional separation between objects of interest, but still have enough class support coverage to maintain robustness. By construction via the greedy approach, one expects that the first cover balls selected to be the “most informative,” since the construction process starts by finding the ball which contains the most in-class points while admitting a fixed number of other-class points (as dictated by $\beta * n_1$). When fewer balls are retained in the cover, fewer in-class samples are used to characterize the target. Therefore, one might expect a degradation in classification performance for values of α at both extremes (*i.e.* close to 1 or close to 0). For α close to zero, most in-class samples are retained, including those that fall in mixed-class regions. As α increases to one (and only a few in-class samples are retained in regions exhibiting high class purity), one would expect an increase in the variance (due to limited sampling) associated with the cover selected by the CCCD process.

2.A.1.1.4. Henze-Penrose divergence

Assume we are given two independent \mathbf{R}^d -valued samples $\mathbf{X}_1, \dots, \mathbf{X}_{n_0}$ and $\mathbf{Y}_1, \dots, \mathbf{Y}_{n_1}$, with the distribution of \mathbf{X}_i having the unknown density $f(\mathbf{x})$ and the distribution of \mathbf{Y}_i having the unknown density $g(\mathbf{x})$. The multivariate two-sample problem is to test the hypothesis $\{H_0: f = g\}$ versus the general alternative $\{H_0: f \neq g\}$. For the multivariate case, several approximate distribution-free methods are known. In this work we use the Henze-Penrose Divergence (HPD), which is an extension of the Friedman-Rafsky statistic. HPD is estimated using information from a minimal spanning tree (MST) computed over the pooled data set $\{\mathbf{X}_1 \dots \mathbf{m} \cup \mathbf{Y}_1 \dots \mathbf{n}\}$. From the MST, remove all edges whose defining nodes originate from different classes. The Friedman-Rafsky test statistic

$R_{m,n}$ is given by the number of resulting disjoint subtrees. The HPD $h(f(\mathbf{x}), g(\mathbf{x}), p)$ is estimated using:

$$h(f, g, p) = \int \frac{p^2 f^2(\mathbf{x}) + (1-p)^2 g^2(\mathbf{x})}{pf(\mathbf{x}) + (1-p)g(\mathbf{x})} d\mathbf{x} \cong 1 - \frac{R_{m,n}}{m+n}, \quad (1)$$

in which $p = m/(m+n)$. In (1), it is required that as $(m \rightarrow \infty)$ and $(n \rightarrow \infty)$, the ratio p remains bounded between 0 and 1. The HPD h has the convenient property that as n and m are increased, the statistic does not change its central value but becomes asymptotically more accurate. The inequality $h(f, g, p) \geq h(f, f, p) = p^2 + (1-p)^2$ is strict for densities f and g that differ on a set of positive measure. For the case $n = m$ (as we use in our analysis), $h = 0.5$ implies that the densities f and g are drawn from the same underlying distribution. As the densities f and g are increasingly separated (less overlap), h increases from 0.5. As the densities f and g become completely separated (disjoint, no overlap), h attains its maximum value $h_{\max} = 1$.

2.A.1.2. Methodology

Our overall goals are to investigate training and testing divergence as a function of α , for some values of β , for two sample sizes. This should provide a basis for better understanding the CCCD approach and for optimizing parameter selection for a particular dataset. We will do this for two two-target examples – in one case, the two targets are known to be dissimilar (target labels T_1, T_2), and in the other case the two targets are known to be similar (target labels T_2, T_2^* are variants of the same target).

For each of three training targets (T_1, T_2 and T_2^*), we generate $\sim 50,000$ evenly-spaced surface vertices by applying ray-tracing-based rendering to CAD models. A spin map is formed for each vertex. The 10×10 spin images are formed using support cylinder radius = support cylinder height = 1.0 m, and support angle = 180° . Each 10×10 spin image is raster-scanned into a 100-dimensional spin-image vector. For training, subsample from $\sim 50,000$ spin-image vectors to $n = 500$ or $n = 2000$ spin-image vectors, using uniform random draw. For testing, subsample from $\sim 50,000$ spin-image vectors to $n = 500$ or $n = 2000$ spin-image vectors, using uniform random draw, having already excluded the $n = 500/2000$ spin-image vectors used for training.

2.A.1.3. Experiments

We apply the CCCD algorithm, with $\beta = \{0.00, 0.01, 0.05\}$, varying $0 \leq \alpha < 1$, for $n = 500$ and $n = 2000$. Two sets of experiments are done: similar targets (T_2 and T_2^*), and dissimilar targets (T_1 and T_2). For each value of α , we repeat the training process $N_{train} = 3$ times, each time with a different n -element subsample drawn from the full spin-image stack. For each experimental realization of the covers \mathbf{C}_0 and \mathbf{C}_1 , we form the joint cover $\mathbf{C} = \mathbf{C}_0 \cup \mathbf{C}_1$. Note that for $\beta = 0$, by construction $\mathbf{C}_0 \cap \mathbf{C}_1 = \{\}$. We estimate the pairwise HPDs $h_{train}(\mathbf{X}_{train}(\mathbf{C}), \mathbf{Y}_{train}(\mathbf{C}))$ between the class-0 training samples $\mathbf{X}_{train}(\mathbf{C})$ that fall within \mathbf{C} and the class-1 training samples $\mathbf{Y}_{train}(\mathbf{C})$ that fall within \mathbf{C} . It is expected that h_{train} is an increasing function of α , as higher values of α correspond to fewer samples from overlapping class-specific support. We compute the mean and standard deviation of the N_{train} measurements of $h_{train}(\mathbf{X}_{\mathbf{C}}, \mathbf{Y}_{\mathbf{C}})$ at each value of α .

Next, for each N_{train} run, we subsample the spin-image stacks (exclusive of samples used for training) $N_{test} = 3$ times, and determine which test samples fall within \mathbf{C} . We then estimate the HPD $h_{test}(\mathbf{X}_{test}(\mathbf{C}), \mathbf{Y}_{test}(\mathbf{C}))$ between the class-0 testing samples

$\mathbf{X}_{\text{test}}(\mathbf{C})$ that fall within \mathbf{C} and the class-1 testing samples $\mathbf{Y}_{\text{test}}(\mathbf{C})$ that fall within \mathbf{C} . We also estimate the train / test cross-divergences $h_{00}(\mathbf{X}_{\text{train}}(\mathbf{C}), \mathbf{X}_{\text{test}}(\mathbf{C}))$, $h_{01}(\mathbf{X}_{\text{train}}(\mathbf{C}), \mathbf{Y}_{\text{test}}(\mathbf{C}))$, $h_{10}(\mathbf{Y}_{\text{train}}(\mathbf{C}), \mathbf{X}_{\text{test}}(\mathbf{C}))$ and $h_{11}(\mathbf{Y}_{\text{train}}(\mathbf{C}), \mathbf{Y}_{\text{test}}(\mathbf{C}))$. We compute the mean and standard deviation of the $N_{\text{train}} * N_{\text{test}} = 9$ measurements of h_{test} and the four train / test cross-divergences at each value of α .

2.A.1.4. Results and Discussion

The HPD h is approximated asymptotically by a linear transformation of the Friedman-Rafsky test statistic $R_{m,n}$. For the two scenarios (similar targets: T2*, T2; dissimilar targets: T1, T2), we estimate h as a function of sample size, with sample size varying from 100 to 2000. The estimates are calculated 10 times using randomly selected sample sets, and the mean and standard deviation of the estimates for h are shown in Figure 3. As expected, the dissimilar targets have greater divergence than the similar targets: $h(T1, T2) > h(T2^*, T2)$. It is noted that with the sample sizes used in this work, $n = 500/2000$, our estimates for the HPD h have not yet reached their asymptotic value, but the curves appear to be flattening at $n = 2000$.

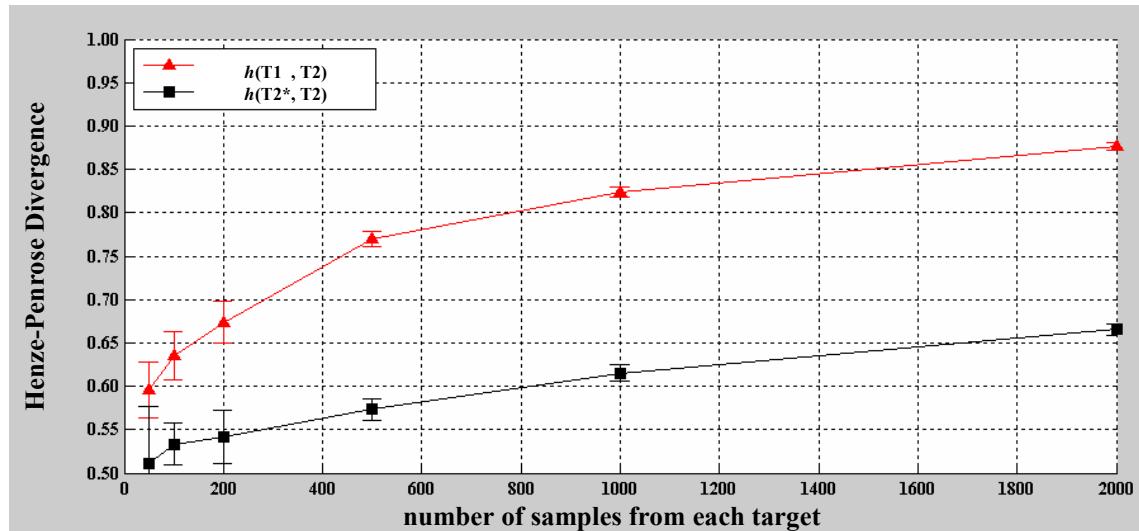


Figure 4: Estimates over 10 runs of HPD as a function of sample size for dissimilar targets, $h(T1, T2)$ and for similar targets, $h(T2^*, T2)$

In Tables 1 – 3, we show the mean and standard deviation of the number of class cover balls for $\beta = \{0, 0.01, 0.05\}$, respectively. For the similar targets, which are expected to have substantially greater support overlap than the dissimilar targets, more balls are required to form the cover. As β increases, fewer balls are required to form the cover. This is expected, as the cover balls need a larger radius to encompass $(\beta * n_1)$ samples. Having a larger radius, more class-0 samples are covered by each ball, so fewer class-cover balls are needed overall. The ratio of balls required for similar targets to balls required for dissimilar targets decreases as β increases.

Table 1: Balls in class covers, mean and standard deviation over 10 runs, for $n=2000$.

CCCD parameter values: $\alpha = \beta = 0$

	Target T1	Target T2	Target T2*	Total Balls
Dissimilar Targets	522 ± 9	371 ± 6		893 ± 12

Similar Targets		832 ± 10	892 ± 15	1724 ± 17
-----------------	--	--------------	--------------	---------------

Table 2: Balls in class covers, mean and standard deviation over 10 runs, for $n=2000$.
 CCCD parameter values: $\alpha = 0$; $\beta = 0.01$

	Target T1	Target T2	Target T2*	Total Balls
Dissimilar Targets	141 ± 4	94 ± 3		235 ± 5
Similar Targets		143 ± 4	162 ± 4	305 ± 5

Table 3: Balls in class covers, mean and standard deviation over 10 runs, for $n=2000$.
 CCCD parameter values: $\alpha = 0$; $\beta = 0.05$

	Target T1	Target T2	Target T2*	Total Balls
Dissimilar Targets	48 ± 3	30 ± 1		78 ± 3
Similar Targets		42 ± 1	48 ± 2	89 ± 2

In Figures 5 – 7, we show histograms and scatterplots of the ball population and radii, for a single random training set, for $\beta = \{0, 0.01, 0.05\}$, respectively. Again, since the similar targets (Figures 5 – 7, bottom row) have more overlapping support, the balls typically have smaller membership and radii than the cover balls for the similar targets (Figures 5 – 7, top row).

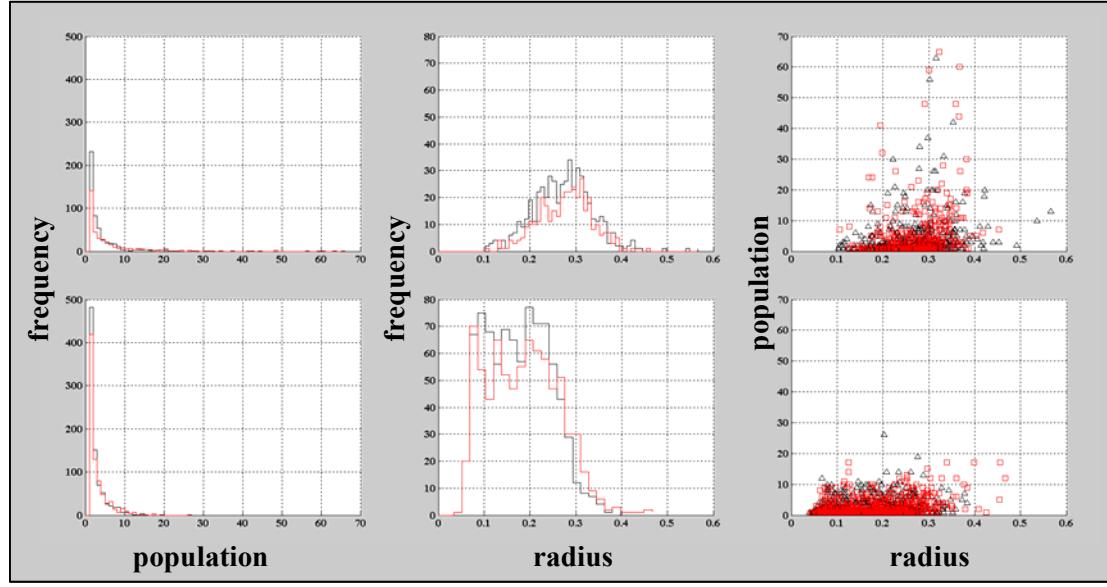


Figure 5: Histograms and scatterplots of population and radii of balls in class covers for dissimilar targets (top row) and similar targets (bottom row), with $n = 2000$. CCCD parameter values: $\alpha = \beta = 0$

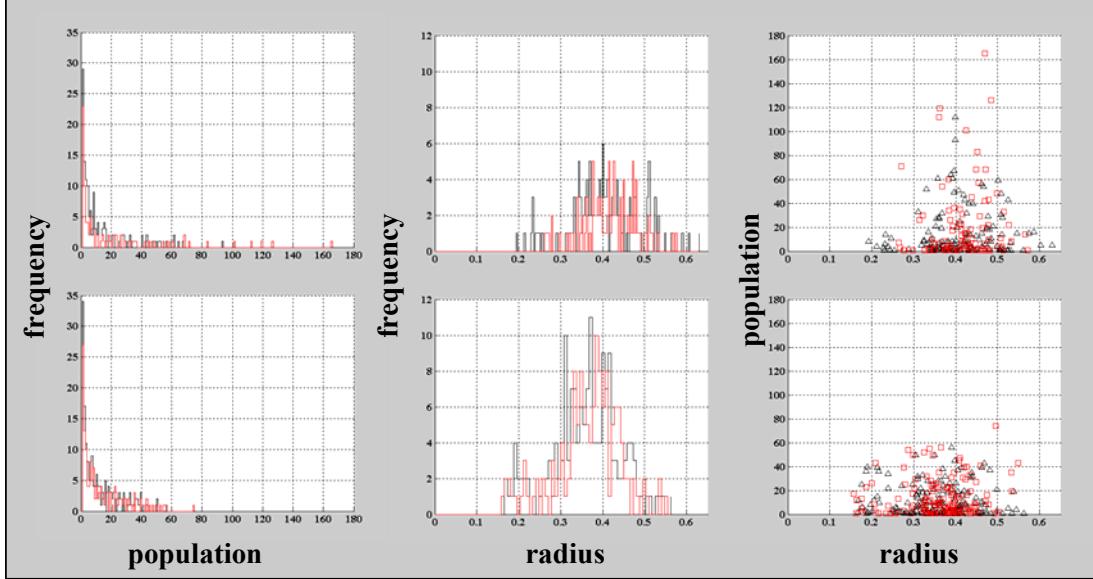


Figure 6: Histograms and scatterplots of population and radii of balls in class covers for dissimilar targets (top row) and similar targets (bottom row), with $n = 2000$. CCCD parameter values: $\alpha = 0$, $\beta = 0.01$

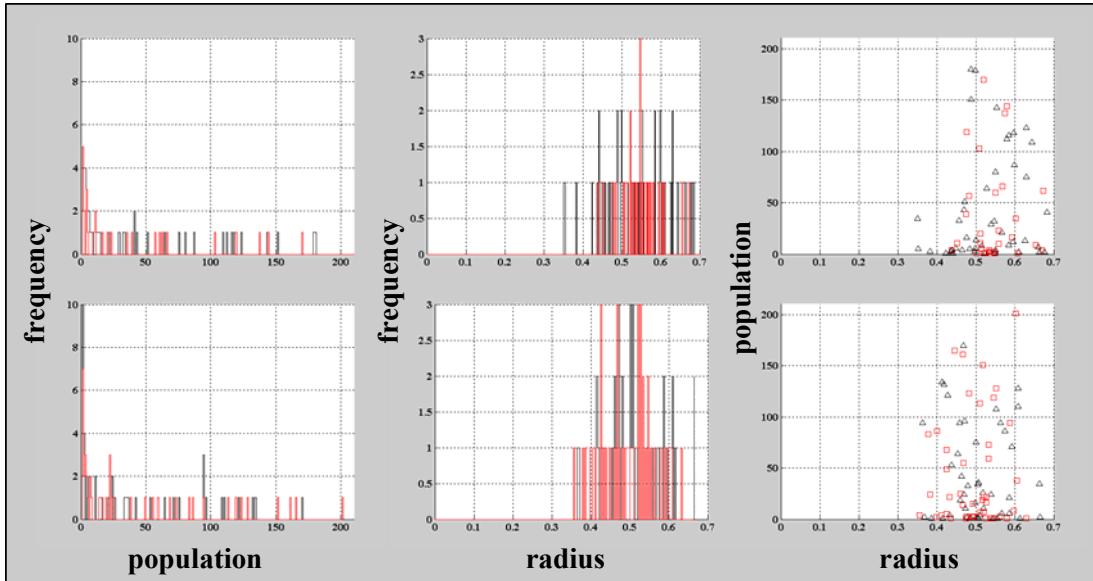


Figure 7: Histograms and scatterplots of population and radii of balls in class covers for dissimilar targets (top row) and similar targets (bottom row), with $n = 2000$. CCCD parameter values: $\alpha = 0$, $\beta = 0.05$

In Figure 7, with $\beta = 0.05$, the plots for the dissimilar targets (top row) are similar to the plots for the similar targets (bottom row). This is in contrast to Figure 5: for $\beta = 0$, the plots for the dissimilar targets are significantly different from the plots for the similar targets. This is understood as follows: as β increases, each ball contains more of the non-target class, so the distinction between dissimilar targets and similar targets is lost – in both cases, the class-specific covers contain an increasing amount of other-class samples. Note that increasing α corresponds removing balls from the cover, starting with the balls with the smallest population. Thus as α approaches 1, we are keeping only the ball(s)

ISP Phase II (Contract N00014-04-C-0437)
Final Technical Progress Report (CDRL A004 No. 1)

having the largest population and the highest ratio of within-class samples to other-class samples.

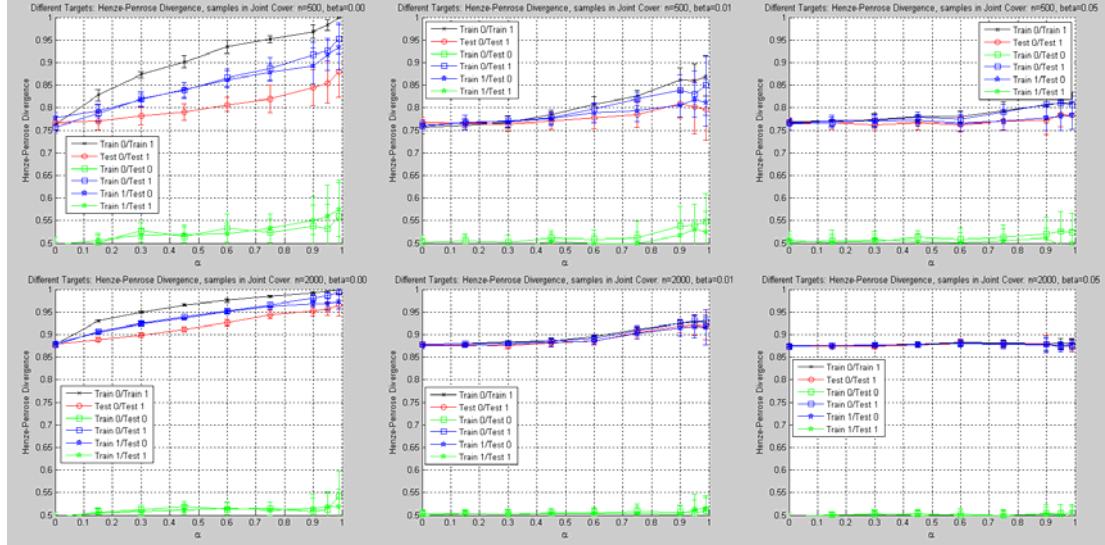


Figure 8: For different targets, estimated divergence as a function of α

In Figure 8 and Figure 9, we show divergence estimates for a pair of different targets and a pair of similar targets, respectively. In each subplot, the divergence is measured between groups of points that fall within the joint cover.

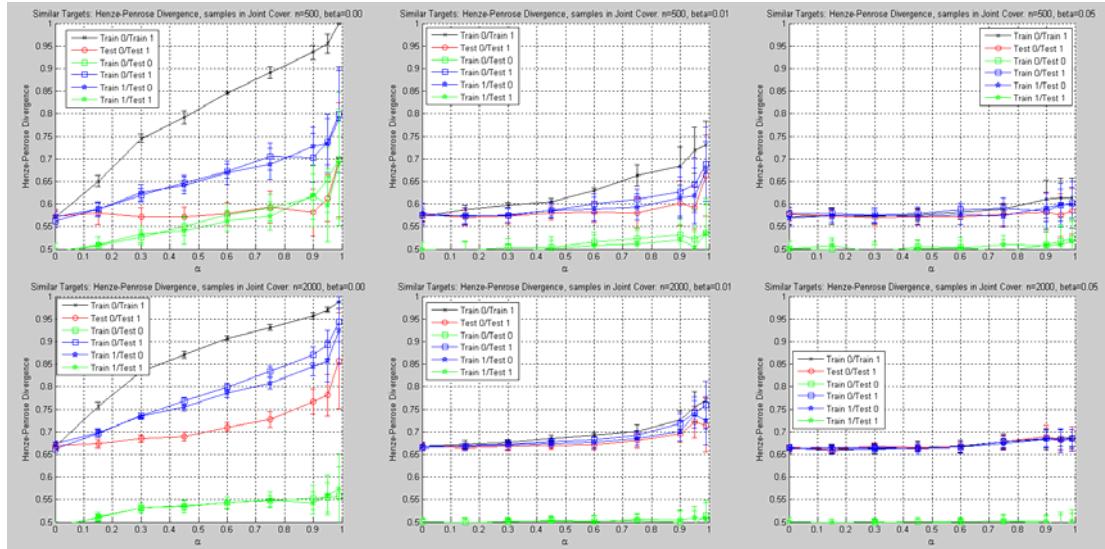


Figure 9: For similar targets, estimated divergence as a function of α

In Figure 8 and Figure 9, the divergence values at $\alpha = 0$ (all data used) are consistent with the $n = 500$ and $n = 2000$ divergence values given in Figure 4. As expected, the training divergence h_{train} is monotonically increasing with α – as fewer balls of higher class population are used, there should be less spatial overlap between inter-class samples. In each of the subplots in Figure 7 and Figure 8, the two green curves represent the train / test cross-divergences $h_{00}(\mathbf{X}_{\text{train}}(\mathbf{C}), \mathbf{X}_{\text{test}}(\mathbf{C}))$ and $h_{11}(\mathbf{Y}_{\text{train}}(\mathbf{C}), \mathbf{Y}_{\text{test}}(\mathbf{C}))$. Since the samples are drawn from the same distribution, we expect $h_{00} \sim h_{11} \sim$

0.5. This is generally the case, except for the $\beta = 0$ subplots (left-hand column in both Figure 8 and Figure 9). Also in the $\beta = 0$ subplots, there is a substantial deviation between the training divergence h_{train} (black curve) and the testing divergence h_{test} (red curve). As $\alpha \rightarrow 1$ we can achieve $h_{\text{train}} \sim 1$, but the h_{test} values remain lower, particularly in the $n = 500$ cases (upper row in Figure 8 and Figure 9). These observations imply that we are overfitting the data, particularly for $n = 500$. The overfitting problem generally worsens as α increases (fewer and fewer training samples used for forming cover).

In the $\beta = 0.05$ (right-hand column) subplots in both Figure 8 and Figure 9, the divergence curves are fairly flat – we appear to be undertraining. There is little separation between the h_{train} and h_{test} values, but neither increases with α . In the $\beta = 0.01$ (middle column) subplots in both Figure 8 and Figure 9, the divergence curves are increasing as α increases, with low variance until $\alpha \sim 0.9$. There is substantially less separation between the h_{train} and h_{test} curves for $n = 2000$, than for $n = 500$. These observations, taken in conjunction with the total number of class cover balls listed in Table 1, Table 2 and Table 3, imply that for both the different target and similar target cases, over the parameter values tested, we should use $\alpha = 0.9+$, $\beta = 0.0+$ and $n = 2000+$ for providing high class separability while maintaining low representation complexity.

For two two-class cases (different targets or similar targets), we have demonstrated a methodology for selecting high-dimensional ($d = 100$) feature space regions (class cover balls) that strike a balance between high class separability and low representational complexity.

2.A.2. Information Theoretic Approaches for Radar Threat Identification

2.A.2.1 Introduction

A radar system operates by radiating electromagnetic energy into space and detecting the echo signal reflected back to the radar from a target. The reflected energy not only indicates the presence of a target, but by comparing the received echo signal with the signal that was transmitted (matched filtering), the target location can be determined along with other target-related information [13]. A hostile target would naturally like to deny the radar system access to this information, if possible, and thus may employ electronic countermeasures. In particular, noise jamming is the intentional transmission of energy in order to mask the target return and impair the effectiveness of the receiving radar. The receiving, or victim, radar may then employ a counter-countermeasure technique known as “home-on-jamming” designed to track the angle of the jamming signal and reveal the location of the jammer. To prevent discovery of his location, the jammer may employ “angle deception” or terrain-bounce jamming. This could be implemented, for example, by an aircraft flying at a relatively low altitude while transmitting energy toward the ground. This technique effectively presents a false targeting angle to the detecting radar, thereby rendering ineffective the home-on-jamming counter-countermeasure. And so it goes.

Obviously, it is necessary to be able to distinguish between a simple direct path jamming signal and a composite signal consisting of the direct path and a bounce path, since the detecting radar must know whether or not it can believe the target angle it is tracking. Under ISP, we have made some small investigations into the question of whether an information theoretic approach, in particular dimensionality reduction, might be of some use in solving this problem. In the last quarterly report, we described our

experiences with processing simulated data that models noise jamming with the ISOMAP algorithm, then computing the Henze-Penrose divergence test between samples of two different types of signals. The results were, ultimately, disappointing. Thus we turned to a simpler analytical model of the received signal, in an attempt to understand what we were dealing with, and finally we feel that we may have gained a tiny bit of insight into the nature of dimensionality reduction solutions for wave phenomena.

In this, our final installment for the ISP Phase II, we outline our meager results. We will begin by introducing a simple model of received signals, the well known plane wave. Next we compute the dimensionality reduction solution, in closed form, that would be obtained with the multi-dimensional scaling algorithm. We show that two combined signals incident from different directions, but with the same frequency, will always appear as a single source in the dimensionality reduction. Finally, we show that a simple, direct path signal cannot be statistically distinguished from a composite signal consisting of a direct path and a bounce path, if these signals all have the same frequency.

2.A.2.2 The Plane Wave: A Simple but Realistic Model

We adopt as our model of incident electromagnetic energy the well known plane wave solution to the wave equation, which is given by

$$w(\mathbf{x}, t) = A_0 \exp(2\pi i f t) \exp(-i \mathbf{k}^T \mathbf{x}),$$

where A_0 is the amplitude, f is the frequency in cycles per second, and \mathbf{k} is the wave number vector, which has magnitude equal to $2\pi/\lambda$, where λ is the wavelength, related to the frequency by $\lambda f = c$, where c is the wave velocity. Additionally, the wave number vector points in the direction of propagation of the wave.

2.A.2.3 The Array Manifold Vector

We collect N time samples at each of M spatial locations, which may be the separate channels of an antenna array, or perhaps a more disparate collection of sensors, our analysis being amenable to either circumstance. Let the positions of the sensors be denoted by \mathbf{p}_m , for $m = 1, 2, \dots, M$. The quantity collected at the time t_n by the m th sensor is

$$w(\mathbf{p}_m, t_n) = A_0 \exp(2\pi i f t_n) \exp(-i \mathbf{k}^T \mathbf{p}_m).$$

Let us arrange these measurements into a vector, which we denote by $\mathbf{s}(t_n)$, defined as

$$\mathbf{s}(t_n) = \frac{1}{\sqrt{M}} \begin{bmatrix} w(\mathbf{p}_1, t_n) \\ \vdots \\ w(\mathbf{p}_M, t_n) \end{bmatrix} = A_0 \exp(2\pi i f t_n) \mathbf{v},$$

where we have defined the following “array manifold vector” (see, *e.g.* [14]),

$$\mathbf{v} = \frac{1}{\sqrt{M}} \begin{bmatrix} \exp(-i \mathbf{k}^T \mathbf{p}_1) \\ \vdots \\ \exp(-i \mathbf{k}^T \mathbf{p}_M) \end{bmatrix}.$$

Note that the array manifold vector is normalized so that $\mathbf{v}^H \mathbf{v} = 1$.

2.A.2.4 Dimensionality Reduction

Let us use the multi-dimensional scaling algorithm (MDS) to assign Cartesian coordinates to the collected samples $\mathbf{s}(t_n)$, $n = 1, \dots, N$. This algorithm may be considered as the prototype of a dimensionality reduction procedure, and probably gives optimal results for linear problems. We first compute the squared pair-wise distances, which are given by

$$\begin{aligned}
 (1) \quad D_{mn}^2 &= \|\mathbf{s}_m - \mathbf{s}_n\|^2 \\
 &= (\mathbf{s}_m - \mathbf{s}_n)^H (\mathbf{s}_m - \mathbf{s}_n) \\
 &= \mathbf{s}_m^H \mathbf{s}_m + \mathbf{s}_n^H \mathbf{s}_n - 2 \operatorname{Re}\{\mathbf{s}_m^H \mathbf{s}_n\} \\
 &= 2A_0^2 [1 - \cos 2\pi f(t_m - t_n)].
 \end{aligned}$$

The goal of the MDS algorithm is to find a set of points in Cartesian space such that the pair-wise distances among these points are equal to the distances that were given to the algorithm. It turns out that, in this simple case, we can explicitly provide just such a set of points.

Define the points $\mathbf{x}_n = (x_n, y_n) \in \mathfrak{R}^2$, for $n = 1, \dots, N$, such that

$$(2) \quad x_n = A_0 \cos 2\pi f t_n, \quad y_n = A_0 \sin 2\pi f t_n.$$

Now let us compute

$$\begin{aligned}
 \|\mathbf{x}_m - \mathbf{x}_n\|^2 &= (x_m - x_n)^2 + (y_m - y_n)^2 \\
 &= (x_m^2 + y_m^2) + (x_n^2 + y_n^2) - 2(x_m x_n + y_m y_n) \\
 &= 2A_0^2 - 2A_0^2 \cos 2\pi f(t_m - t_n),
 \end{aligned}$$

which clearly agrees with (1), and proves that (2) is the dimensionality reduction for this data. This solution is unique only up to a translation and rotation, hence to see perfect agreement between a computed solution and our hand-derived solution would most likely require “centering” the hand solution to have zero mean then computing a rotation matrix that maps one solution onto the other.

Note that (2) correctly captures the amplitude and frequency of the incident wave, but all information about the direction of the source in relation to the sensor array is lost. The direction information was contained in the array manifold vector and, perhaps unfortunately, by converting the measurements to distances we have effectively removed it. Interestingly, this also implies that the result is independent of the number of sensors; we would expect to obtain the same result with one sensor as with one hundred. Finally note that the dimension for a single (complex) plane wave source is two.

2.A.2.5 The Case of Two Plane Wave Sources

Now let us consider the case of a composite signal, consisting of two plane waves, with differing angles of incidence. We assume different amplitudes, and a time lag in one of the signals with respect to the other, since it is not reasonable to expect that the two signals will arrive at the sensors exactly in phase with each other, or with the same

amplitudes. However we assume the same frequencies in both waves. Since it is our intention to model the contrast between a direct and a reflected path, and since the reflection may change amplitude, or polarization, but not the frequency, unless the surface off which the wave was reflected was moving, but ours is not moving, therefore we assume that the direct path and reflected wave have the same frequency. This is also convenient in that, under the assumption of same frequency, we can give an explicit solution to the dimensionality reduction in this case also.

Let the time samples collected at the sensors have the following form,

$$(3) \quad \mathbf{s}(t_n) = A_1 \exp(2\pi f t_n) \mathbf{v}_1 + A_2 \exp(2\pi f(t_n + \tau_0)) \mathbf{v}_2,$$

where \mathbf{v}_1 and \mathbf{v}_2 are two different array manifold vectors. Expression (3) is our model of a “composite” signal, consisting of one plane wave with wave number vector \mathbf{k}_1 , and a second one with a different wave number vector \mathbf{k}_2 , which has the same magnitude but a different direction than \mathbf{k}_1 , i.e. the second wave is incident from a different direction. The second plane wave in (3) also has different amplitude and is time-delayed with respect to the first although, as explained above, the two waves have identical frequency.

We will not reproduce the tedious calculations here, but ask the gentle reader to take our word, that the dimensionality reduction of samples in (3) is given by

$$(4) \quad x_n = \Delta \cos 2\pi f t_n, \quad y_n = \Delta \sin 2\pi f t_n,$$

where the amplitude of the solution is given by

$$(5) \quad \Delta^2 = A_1^2 + A_2^2 + 2A_1 A_2 [C_{12} \cos 2\pi f \tau_0 - S_{12} \sin 2\pi f \tau_0],$$

with

$$(6) \quad C_{12} = \frac{1}{M} \sum_{j=1}^M \cos[(\mathbf{k}_1 - \mathbf{k}_2)^T \mathbf{p}_j], \quad S_{12} = \frac{1}{M} \sum_{j=1}^M \sin[(\mathbf{k}_1 - \mathbf{k}_2)^T \mathbf{p}_j].$$

Note that the time lag τ_0 appears in the amplitude term (5), as do the two individual amplitudes, also the array manifold vectors make their influence felt through the terms in (6), though note it is only the difference between the two vectors that appears. It seems clear that any information which we may or may not be able to extract from the dimensionality reduction (4) is contained in the amplitude term (5). At least, the situation does not seem to be entirely hopeless, since the amplitude is a function of the various parameters.

Now the following question arises. Could we, with the aid of dimensionality reduction followed by statistical discrimination, distinguish between the expressions (2) and (4)? We will give the results of attempting this experiment in the following section.

2.A.2.6 Numerical Validation of these Results

Let us first demonstrate the veracity of our closed-form solution in equation (2). We chose parameters convenient for plotting. Throughout these experiments we used $c = 1$, $\lambda = 1/2$, and $f = 2$. In order that we may not be accused of choosing sampling times which are too regular, we drew sampling times at random from a uniform distribution, stretched over many periods of the incoming signal, then sorted into

ascending order. We used one hundred time samples. There were five sensors in a planar configuration.

On the left-hand of Figure 10 we show a plot of the singular values of the double-centered matrix of squared pair-wise distances, the singular value decomposition of which provides the embedding coordinates computed by the MDS algorithm. For this plot we set $A_0 = 1$. We see clearly that the dimension is two, as predicted by our formula. The singular values are not of equal magnitude, but it is unlikely they will ever be so, and it is a fact they are not equal to the amplitudes of the closed form solution in (2), i.e. they are not equal to A_0 . Rather, it is the product of the singular value and the singular vector that is “equivalent” (up to translation and rotation) to the closed form solution in (2).

On the right-hand of Figure 10, we show the differences between embedding coordinates computed with the MDS algorithm and embedding coordinates computed with our closed-form expression (2), after rotation of the latter onto the former. The rotation matrix can be obtained via a simple least squares procedure, but is of course data dependent. As can be seen in the figure, we have perfect agreement up to the numerical precision in sixty-four bit arithmetic. This gives numerical confirmation that expression (2) is correct.

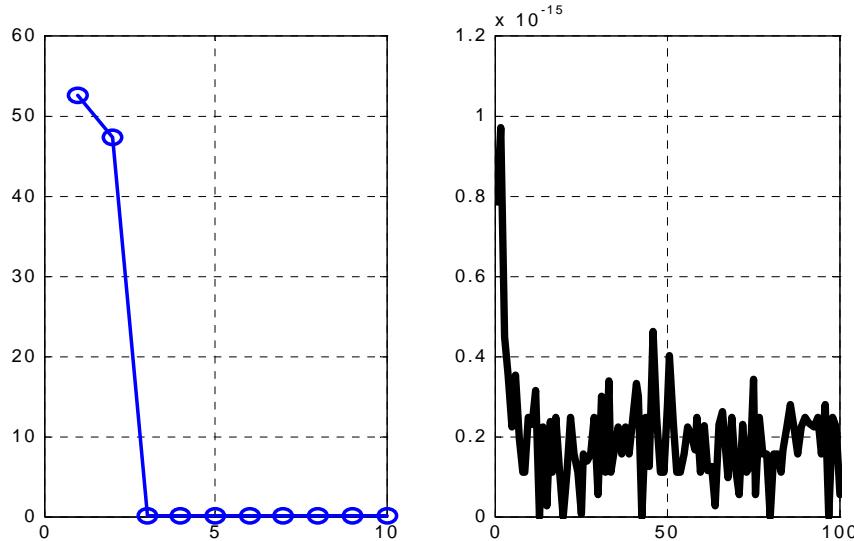


Figure 10: Singular values on the left, and differences between computed and hand solution on the right, for a simple wave.

Let us now turn our attention toward the closed-form expression (4) for a “composite” signal, i.e. consisting of two components from two different directions but with identical frequency. For this experiment we chose $A_1 = 1$, $A_2 = 0.5$, and $\tau_0 = 0.05$. Figure 11 is the analogue of Figure 10. Note especially that, again, the dimension is two as predicted by our closed-form solution, in spite of the fact that the signal consists of two incident waves arriving simultaneously at the sensors. We might expect that each wave would produce two embedding coordinates, for a total dimension of four, but this is not the case. The right-hand side of Figure 11 is numerical confirmation of the correctness of the results given above in equations (4), (5), and (6).

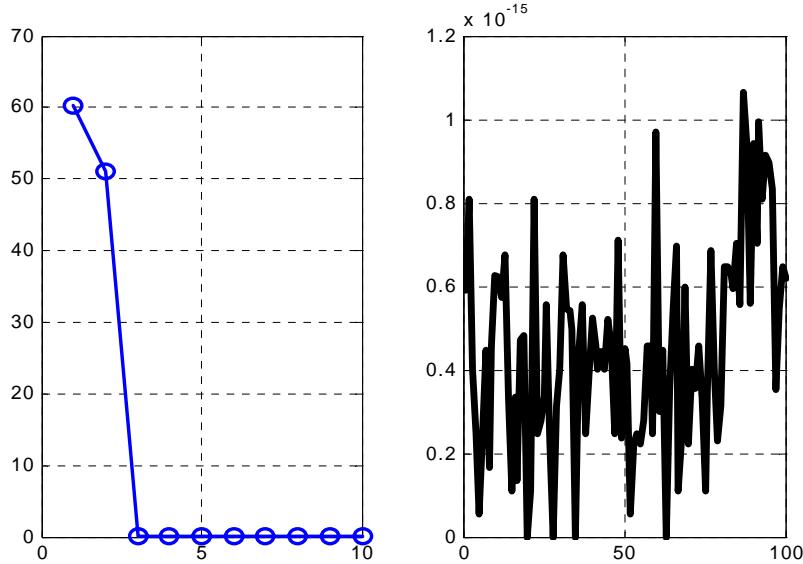


Figure 11: Singular values on the left, and differences between computed and hand solution on the right, for a simple wave

2.A.2.6 Discrimination between Simple and Composite Models

Finally we attempt to see statistically significant differences between our simple model in (1) and our composite model in (3). The procedure is to pack time samples collected with both models into a single array and compute embedding coordinates with MDS. Then we examine the singular values and the embedding coordinates to see what we can see. We will confess at the outset that we do not understand these results particularly well, so we will merely show a couple of examples, then move on to the wrap-up.

For the first experiment, we set $A_0 = 1$ in the expression (1), and set $A_1 = 1, A_2 = 0$ in the expression (3). Thus this experiment compared two simple signals against each other. We used different sequences of randomly generated sample times for the two models. The results are shown in Figure 12. In the left window, we show the singular values. The embedding dimension now is four. Clearly the dimensionality reduction has “seen” two components in the signal.

We show the embedding coordinates for the first two dimensions in the middle window, in blue and red. We apologize for making the plot so small that it is difficult to distinguish the two, but it is easy enough to see that both embedding coordinates are sinusoids, as in expressions (2) and (4), and moreover that these first two have, at least roughly, the same amplitude. The first 100 are the embedding coordinates for the expression (1) and the second 100 are the embedding coordinates for the expression (3).

The embedding coordinates for dimensions three and four are shown in the right window. Judging by the nearness of the first two singular values to each other, and the third and fourth singular values to each other, it seems natural to assume that the first two dimensions belong with the first of the two components, and the third and fourth dimensions with the second of the two components. But let us be careful to avoid saying

“components of the signal,” since it is not entirely clear what these components represent, although they are certainly similar to our expressions in (2) and (4). In fact, from our expressions (2) and (4) we would expect to see the graphs of two sinusoids with the same frequency and amplitude, and that is what we see in the middle and right windows of the plot.

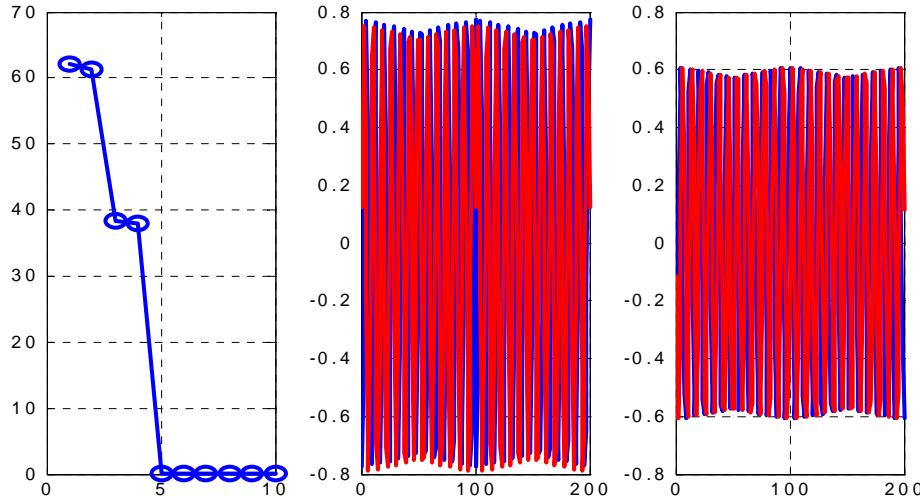


Figure 12: Comparison of two simple signals

We performed a second experiment, setting $A_0 = 1$ in the expression (1), and $A_1 = 1, A_2 = 0.5$ in the expression (3). This experiment compared a simple signal against a composite signal. The results are shown in Figure 13. Note that the singular values are very nearly the same as in Figure 12, but the embedding coordinates have taken on a rather different character. Could this be used as a discriminator? Perhaps, but at the moment we do not know, and there is no more time. This investigation is finished.

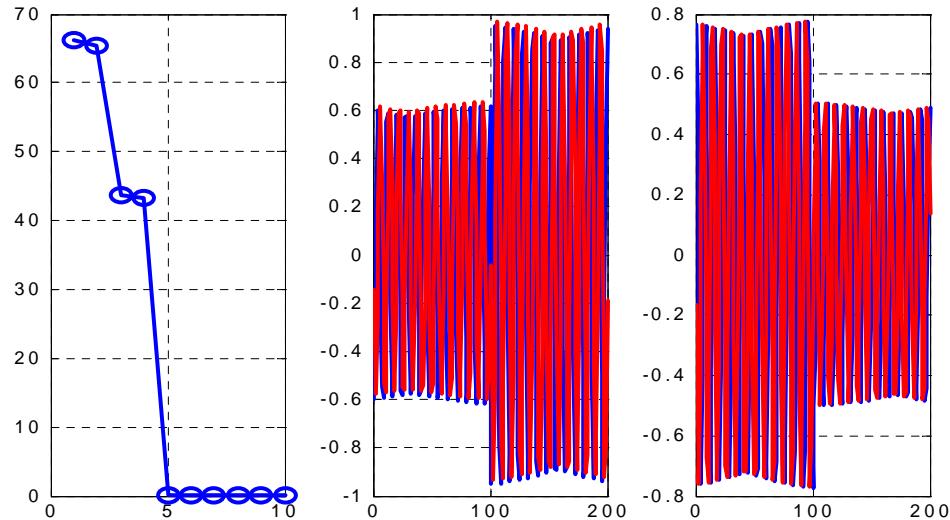


Figure 13: comparison of a simple signal and a composite signal.

2.A.2.7 Analysis of our Composite Model via the Covariance Matrix

A more traditional approach to such problems is to form the covariance matrix of the received signal vectors and compute its eigendecomposition. It is possible to write down an explicit expression for the covariance matrix obtained with our composite model (4). Let us denote the covariance matrix by \mathbf{C} . Then we have

$$\mathbf{C} = A_1^2 \mathbf{v}_1 \mathbf{v}_1^H + A_2^2 \mathbf{v}_2 \mathbf{v}_2^H + A_1 A_2 \left\{ \exp(-2\pi i f \tau_0) \mathbf{v}_1 \mathbf{v}_2^H + \exp(2\pi i f \tau_0) \mathbf{v}_2 \mathbf{v}_1^H \right\}.$$

From this expression it can easily be shown that, for any choice of the array manifold vectors \mathbf{v}_1 and \mathbf{v}_2 , the rank of \mathbf{C} is one. That is, the dimension of the signal subspace is one, not two, as we might have hoped. One way of understanding this is to realize that the two components of the composite signal are almost perfectly correlated. We believe this feature of our model is consistent with the true nature of the problem, since the second component of our composite signal is modeled as a reflection of the first component, thus it is not surprising they are correlated. The same observation probably also explains why the embedding dimension of our composite model was two not four.

2.A.2.8 Conclusions

We have little to add to the foregoing. However, we will make one observation from the viewpoint of computational efficiency. We used a number of sensors equal to M and collected a number of time samples equal to N . Traditional array processing techniques compute the eigendecomposition of a covariance matrix, which has size $M \times M$. Our experiments with dimensionality reduction required us to decompose a distance matrix of size $N \times N$. Since M is usually much smaller than N it follows that nobody in their right mind would choose to compute the eigendecomposition of the distance matrix over the covariance matrix unless the dimensionality reduction offered a significant benefit over the more traditional techniques. So far, we have not seen that it does. However, we cannot positively assert that it doesn't, either.

2.A.3 ASU Technical Progress

2.A.3.1. Introduction

In this section, we summarize the work done at ASU since the last ISP Phase II progress report. This work includes:

- Support of the Raytheon effort to develop the mote-based tracker.
- The integration of the Georgia Tech imager API into the ASU person tracker.
- Application of integer non-linear programming for a constrained non-myopic sensor scheduling problem.

2.A.3.2. Mote-based Tracker

The ASU mote effort included the following components:

- Tracker characterization using one bit detection and energy data,
- Refining the on-mote footstep detector,
- Tuning the particle-filter based tracker.

Tracker Characterization Using One-Bit Detection and Energy Data

As part of the ISP Phase II demo, we compared the performance of target trackers using one-bit detections and received energy measurements; both are collected by a grid of motes with acoustics sensors. The purpose of this comparison was to determine the effect of data reduction at the motes on track accuracy. Particle filter trackers were used,

since they are simple to implement and provide asymptotic optimality as the number of particles grows large. The performance of the two trackers was evaluated using Monte Carlo simulations. Simulation models for the detection and energy data were developed from mote characterization data obtained during the development of the on-mote footstep detector. The motes measure the received acoustic energy from footsteps using the on-board microphone. During the development of the on-mote footstep detector described in previous progress reports, the acoustic sensor was characterized by obtaining received energy values from the motes at target distances varying from 2 to 15 feet. The plot of measured footstep energy as a function of distance with means and confidence intervals is shown in Figure 14.

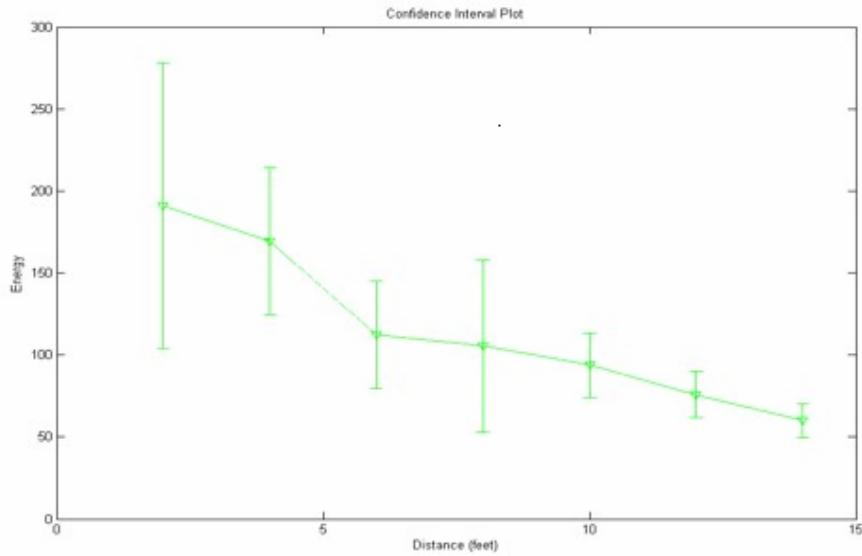


Figure 14: Footstep energy as function of distance: means & confidence intervals

A curve was fit to the energy-distance data; this curve is given by

$$E = \frac{a}{d} + c$$

where the fixed constants were found to be $a = 291.34$ and $c = 61.45$; d is the distance from the sensor measured in feet.

The detection performance of the footstep in the presence of interfering speech was characterized in previously reported work. Some resulting probability of detection and probability of false alarm curves are shown in Figure 15. For the purpose of this investigation, we developed a simple piecewise linear approximation to the probability of detection curve with no interfering speech. This approximation was used to simulate target detections.

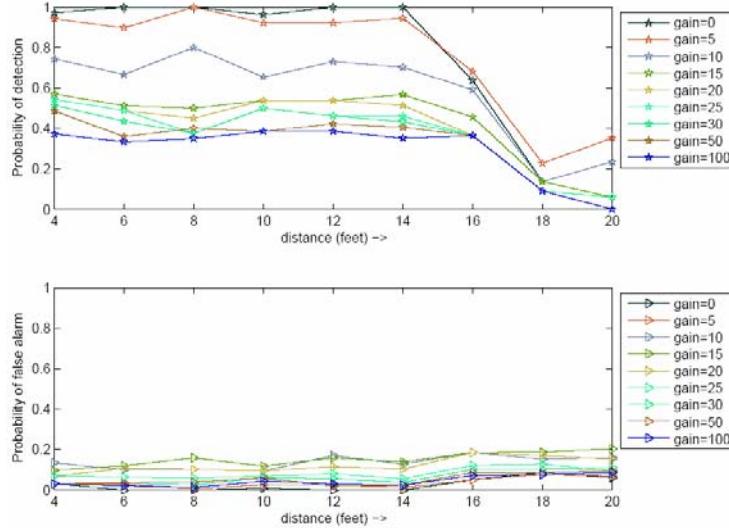


Figure 15: P_D and P_{FA} in the presence of interference

Both the 1 bit (detection-based) and energy trackers were implemented, and each tracker's performance was evaluated. In the simulation, 36 motes were placed in a 6x6 grid with 2 m spacing. The target moved across the mote grid as shown in Figure 16. Target observations are obtained once each second. Both trackers used 4,000 particles. A constant velocity target model is used. Average mean squared error (MSE) was computed for both the trackers over 100 Monte Carlo simulation runs. The tracked trajectory and the tracker MSE over time for the 1 bit tracker and energy tracker is shown in Figures 17 and 18, respectively.

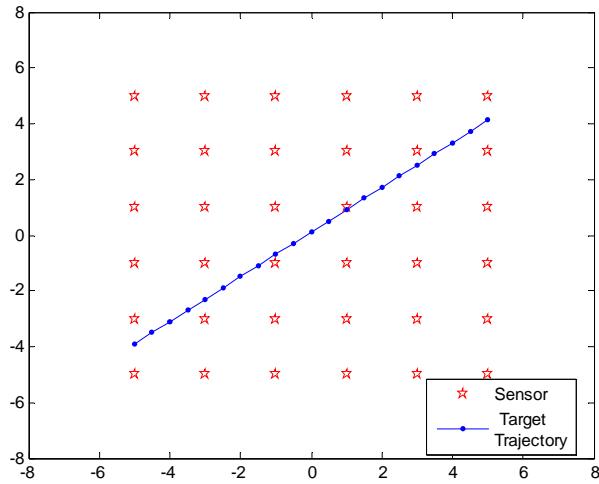


Figure 16: Sensor grid with target trajectory; the grid spacing is in meters.

It can be observed that the MSE for the 1-bit tracker is typically 2-3 times larger than the MSE for the energy tracker. This is because there is significant variation in received energy for footsteps at a given distance, so the relationship between distance and energy is not strongly informative. The probability of detection decreases rapidly at 14-

18 feet, so a dense grid of detectors can localize a target fairly well as long as several sensors are used.

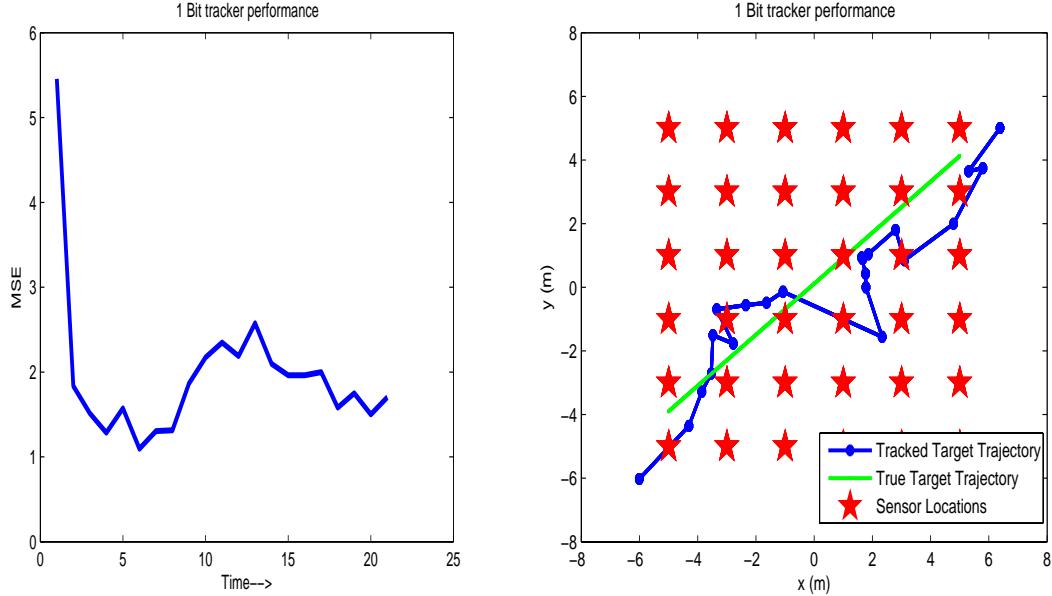


Figure 17: 1-Bit Tracker Performance

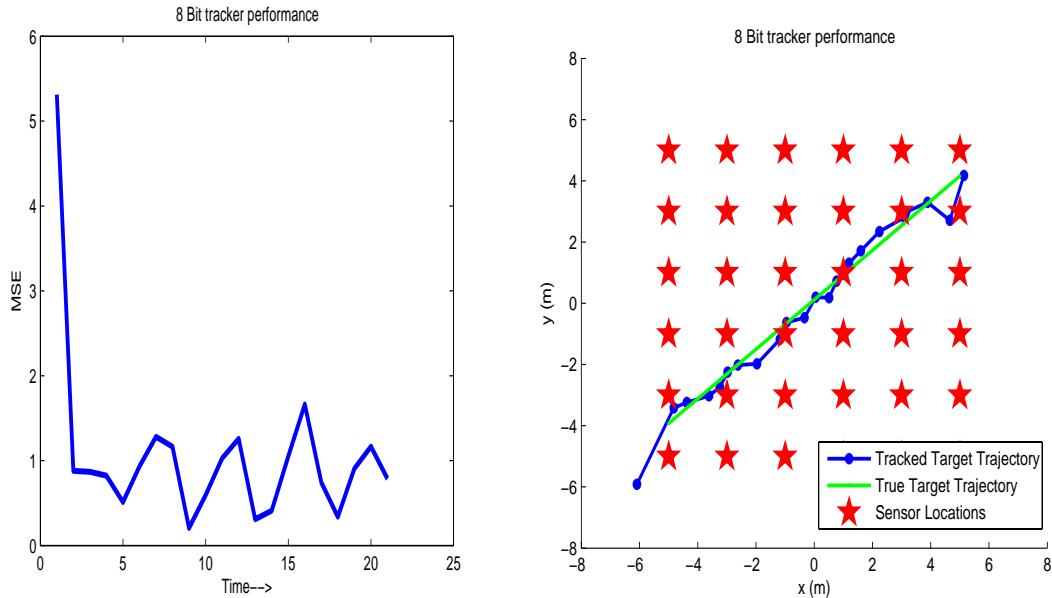


Figure 18: Energy Tracker Performance

Refining the On-Mote Footstep Detector

Occasionally, the footstep detector would give multiple detections in a single footstep. This is due to multiple peaks in the received acoustic energy which may be due to the following reasons:

- a) Heel strikes first, then toe
- b) Reverberation in enclosed space.

Such a situation is demonstrated in Figures 19 and 20. The segment marked in red in Figure 19 is shown zoomed in Figure 20 to show how a single footstep can have multiple peaks and can thereby cause multiple detections.

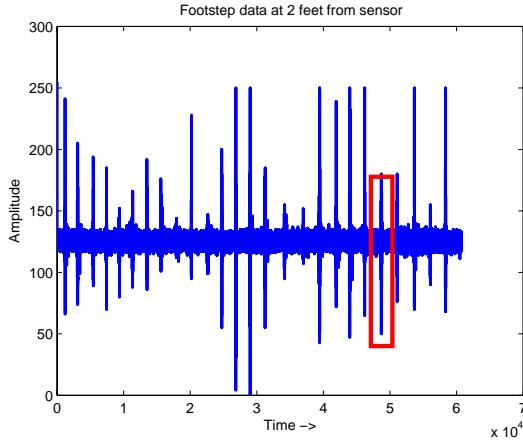


Figure 19: Footstep data at 2 feet from sensor

The footstep detection logic was extended to not record a second detection when two threshold exceedances occur quickly in succession. Instead, a check is performed on whether the energy value remains below a threshold for a certain number of times. This extension reduces the false alarm due to multiple detections.

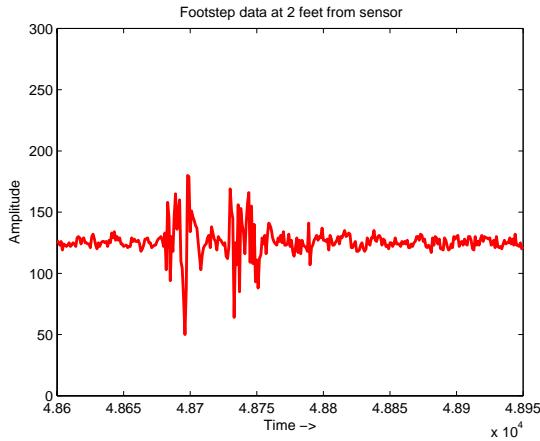


Figure 20: Zoomed version of a single footstep

Tuning the Particle-Filter Based Tracker

For the demonstration, Raytheon has developed a graphical user interface that allows data to be collected from the motes and processed by one of several different trackers. One of the trackers is a particle filter tracker, developed by extending an algorithm originally developed at ASU in 2003. It was observed that this tracker did not give good performance, and so some time was invested in tuning this algorithm for the motes. The complex model for the probability that a detection is returned (*i.e.*, the observation is one) as a function of the target/sensor distance in the original algorithm was replaced by a piece-wise linear function that is parameterized by the following four values as shown in Figure 21:

<i>pd</i>	The probability that a target close to the sensor is detected.
<i>pfa</i>	The probability that the sensor gives a detection and no target is close to the sensor.
<i>pd_dist</i>	The maximum target distance at which the probability of detection is <i>pd</i> .
<i>pfa_dist</i>	The minimum target distance at which the probability of detection is <i>pfa</i> (<i>i.e.</i> , minimum distance at which the target is not likely to be detected).

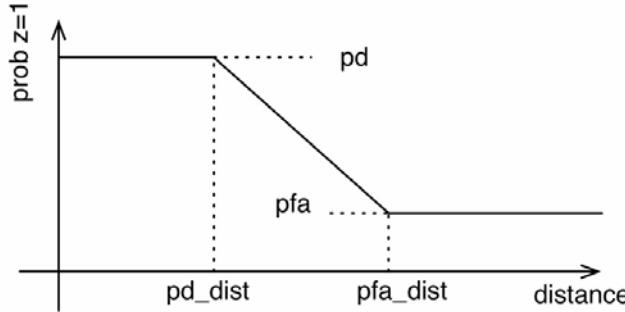


Figure 21: Probability of a detection as a function of sensor/target distance.

In the original tracker, particles that fell outside of the sensor field were not eliminated. If they are not eliminated, then fairly early in the simulation, most of the particles become located outside the sensor field, which makes it impossible for the filter to track the target. The reason this happens is that particles outside the sensor field are neither confirmed nor denied by sensor observations, so most of them survive resampling. The simulation was modified so that particles that fall outside the sensor field were given a weight of zero, which means that they are never resampled.

In the original tracker, the velocities of initial particles were too large; they could have magnitudes up to 60 m/s. The maximum initial velocity was limited to 1 m/s. The process noise matrix did not account for the variable time between observations; when the time between observations is large, the entries in process noise matrix should be larger. It was updated to use the process noise matrix structure for a nearly constant velocity model.

The sensor activation algorithm was modified to use the error variance of the predicted particles, so the sensors are activated at the locations where the target is predicted to be, not at the estimated target location at the time of the last observation. This allowed the factor that is used to compute the radius of the circle inside which sensors are turned on to be decreased.

2.A.3.3. Application Programming Interface Development

In previously reported work, we have developed a particle filter based tracking algorithm for tracking varying number of people from video sequences obtained from an imager. In the current work, we have interfaced this tracker with the ST imager using the image hardware/emulator API supplied by Georgia Tech. The tracking code requires the computation of a weighted sum of pixels from a specified block of the image; the weighting function is either Gaussian or Mexican hat. We have formulated this operation for the matrix operations of the Georgia Tech imager. Computing the sum of the Mexican hat weight requires some post processing of the values read from the imager. On the other

hand, computing the sum of the Gaussian weight can be done entirely by the imager because the Gaussian weight matrix can be expressed as an element wise product of two vectors. The tracker using the imager API was evaluated using several test video sequences to verify that the API integration was correct.

2.A.3.4. Non-myopic Sensor Scheduling

In previously reported work, we have developed myopic sensor scheduling policies for a network of bearing sensors consisting of the regular measurement acquiring sensor nodes and multiple measurement fusing leader nodes (which also hold the target state belief) as shown in Figure 22. In this reporting period, we have extended this work to non-myopic scheduling for the problem that we call the Leader Node Scheduling (LNS) problem. This problem is tracking a target in a distributed sensor network consisting of bearing sensors nodes and leader nodes that fuse the target originated measurements acquired by the sensor nodes. The scheduling problem was formulated as a constrained optimization problem where the sensor usage and communications cost over the planning horizon are optimized subject to tracking error constraints for each planning step. The search for a sensor schedule that obtained a leader node and a subset of sensor nodes at each planning step was then obtained using efficient search strategies. We structure this entire constrained problem as an Integer Non-linear Programming (INLP) and solve it using outer approximation (OA).

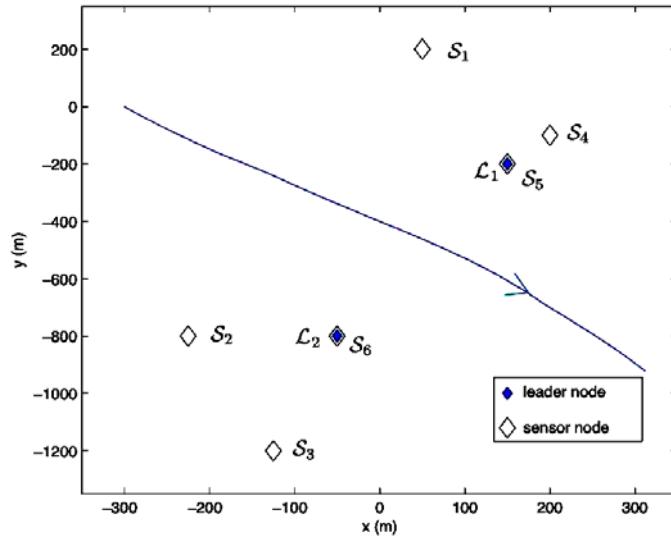


Figure 22: Leader node scheduling problem scenario

The plot of running average energy in Figure 23 for planning horizon lengths $M=1$ and $M=3$ clearly shows the superior performance of the NMSS policies over myopic policies. This superior performance is achieved by transferring the target state belief to another leader node if doing so results in accrual of better sensor energy usage.

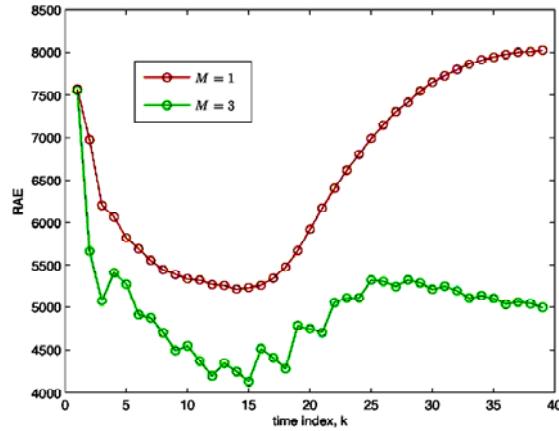


Figure 23: Running average energy (RAE) as a function of time

We have also applied integer non-linear programming to the problem of sensor scheduling for a dense sensor network that consists of acoustic sensor nodes that have a usage and start-up cost; this network has a single fusion center. We call this problem the Central Node Scheduling (CNS) problem. Here we minimize the total predicted tracking error over the M step planning horizon subject to sensor cost constraints. The sensor costs stem from sensor usage and activation costs, where the activation costs are greater than usage costs. We again formulate the problem as an INLP problem and solve it using OA. Figure 24 shows the MSE plot and Figure 25 shows the sensor cost for different scenarios. In these figures, SCC denotes strong cost constraints that incorporate sensor activation costs and WCC denotes weak cost constraints for which the sensor activation cost is set to zero. T denotes the execution horizon length. These plots clearly illustrate the advantage of achieving cost efficient strategies for sensors having start up costs at the expense of only a slightly poorer MSE performance as is seen for $M=10$, $T=3$ for SCC and WCC cases.

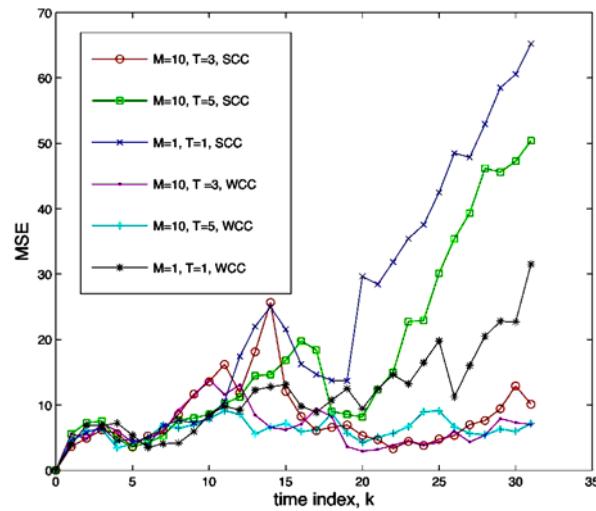


Figure 24: MSE plot for the central node scheduling problem

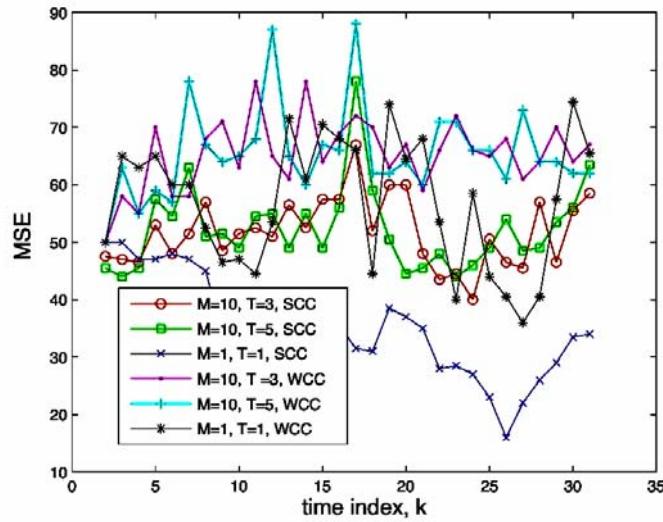


Figure 25: Sensor cost for different scenarios in the central node scheduling problem

2.4.4. Distributed Mote Tracker Support

2.A.4.1. Tracker Test Bed and Test Site

One of the goals of ISP II was to complete a multi-sensor test bed to test on-mote detection algorithms and on-processing-station tracking algorithms. Our test bed (show in operation in Figure 26) is also capable of running multiple tracking algorithms on several processing stations.



Figure 26: Raytheon Distributed Tracking Test Site

In terms of the hardware, the test bed consists of MICA2 Motes equipped with a multi-sensor board, and a Mobile PC as a Processing Station. The software used for the test bed is TinyOS for programming the motes, java for handling the interface between

the serial data of the RS-232 and MATLAB, and MATLAB scripts for the trackers. Referring to Figure 27, each mote has its flash memory partitioned by Deluge in 6 different partitions. Deluge is a UC Berkeley application that formats the flash and programs the remote sensors by epidemic propagation. For our demo four partitions or applications were used: Golden Image, Distributed Weighted Multi-Dimensional Scaling (DWMDS) localization, Beeper Detector, and Footstep Detector. The Golden Image is a safety application so that in the event an application fails to execute, one can always gain control of the remote sensor or mote. The DWMDS is the algorithm used for self-localization provided originally by the University of Michigan; for data this algorithm uses the Receiver Signal Strength (RSS) from each mote. The Beeper Detector and Footstep Detector detect a beeping signal of approximately 4 KHz, and a footstep like signal, respectively. The Beeper Detector was provided by Vanderbilt University, and the Footstep Detector was provided by Arizona State University. Each MICA2 mote transmits its radio signal to a Base Station mote which is connected to a Processing Station via a serial buss (i.e., RS-232). The received data packets from the RS-232 are handled by the Gateway.

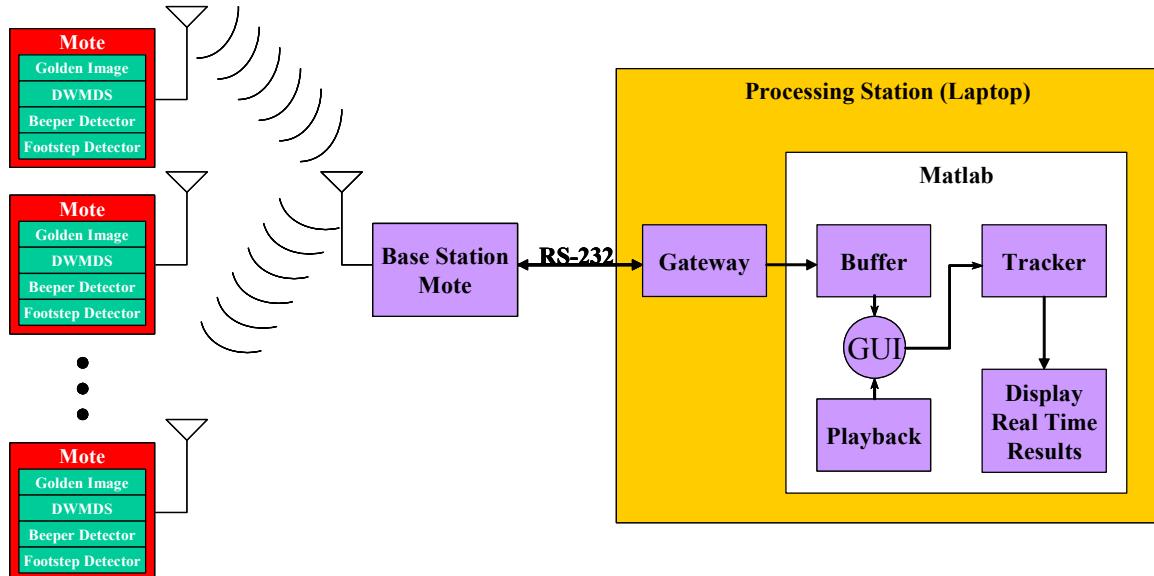


Figure 27: Raytheon ISP Test Bed

The Gateway consists of a *SerialForwarder*, a *Listen* application in java, and a *RemoteControl* application also in java (c.f., Figure 28). The *SerialForwarder* forwards the packets from the serial port to a server port connection so that other programs can communicate with the sensor network via a Base Station. The *Listen* application captures the messages arriving at the Base Station and prints them on the Processing Station (*i.e.*, laptop) screen, and the *RemoteControl* sends commands to the motes to reboot into a different partition or application, or change application parameters.

The parsed data from the Gateway is collected into Buffers or Files where they are read real-time by the MATLAB Graphical User Interface (GUI) (c.f., Figure 29). The GUI is launched at the MATLAB command prompt by typing ISPgui. The user using this GUI can control the functionality of the ISP Multi-Sensor Test bed.

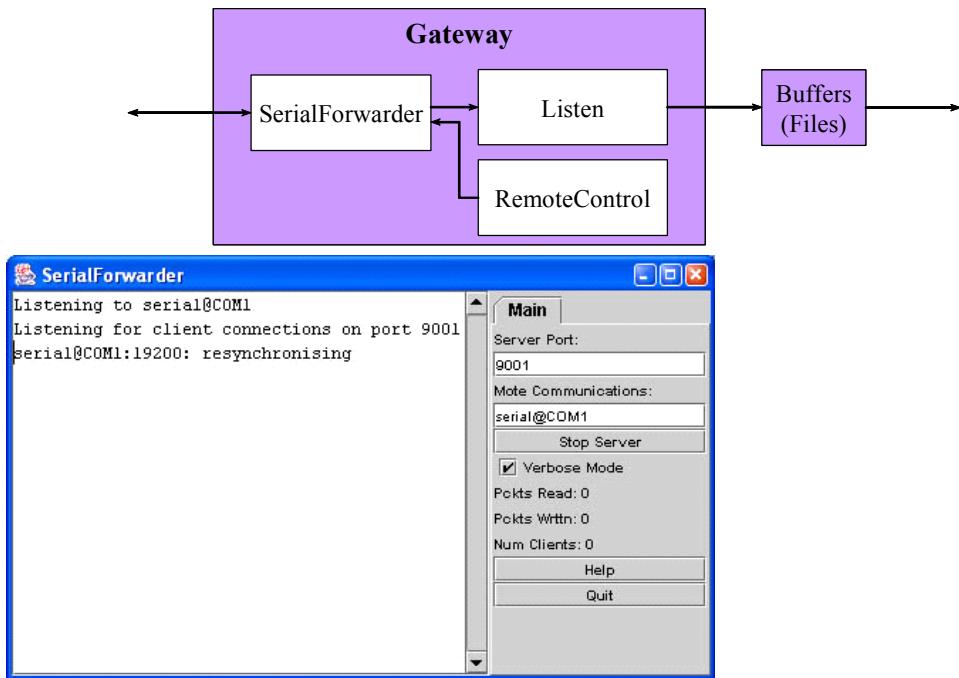


Figure 28: Gateway Overview

The user through the GUI can give a descriptive name to his/her experiment which become part of filenames for the movie and playback file. Furthermore, in Live Mode the user can either the Beeper Detector or the Footstep Detector.

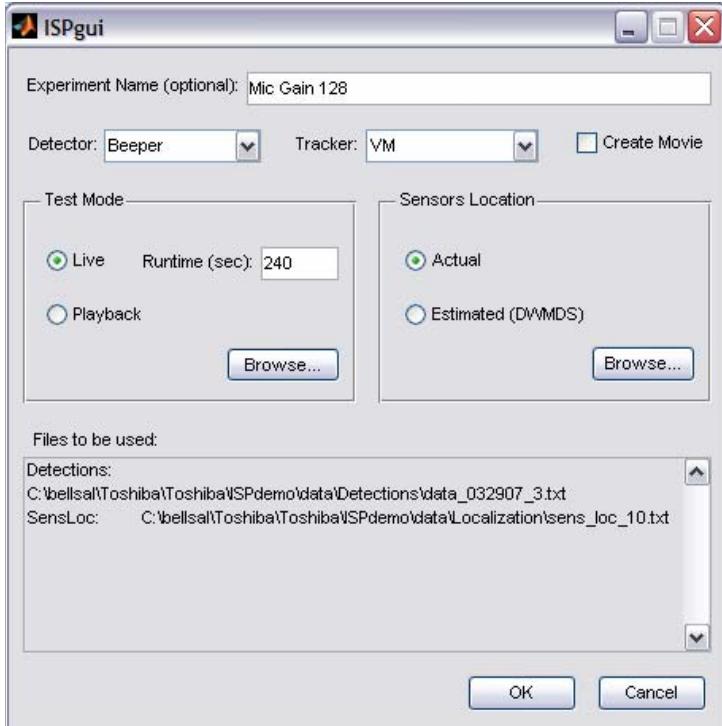


Figure 29: ISP Phase II Test Bed MATLAB GUI

Also, the user has four choices for the trackers: Virtual Measurements (VM), Unscented Kalman Filter (UKF), Particle Filter (PF), and None. The VM and the UKF were both provided by the University of Melbourne, and the PF was provided by ASU. The None option is used if the user wants only to collect the data from the detectors, and then run the tracker offline through the playback files. The Create Movie option can be chosen if one desires to save the results of the trackers in a movie file (*i.e.*, AVI file). The Test Mode has two mutually exclusive options: Live or Playback. The Live option can only be selected in the presence of the MICA2 motes. Also, this option reads the incoming detections real-time, time stamps the detections and saves them into a Playback file. Furthermore, when the Live option is selected, the user can enter the amount of Runtime in seconds that he/she wishes to run the tracker and/or collect data. The Playback option is chosen for playing back offline the detections through the tracker. The Browse button is used so one can choose the detection file or the playback file depending if Live Mode or Playback Mode was chosen.

As for the Sensor Location, one can select either Actual or Estimated. The Actual option refers to the true location of each sensor, while the Estimated option refers to the estimated location of each sensor from localization algorithms such as the DWMDS, RIPS, or the Acoustic Ranging. The Browse button is used to select the file containing the location of the remote sensors. The area at the bottom of the GUI entitled *Filed to be used* (c.f., Figure 29) summarizes the selections that the user has selected. Once the OK button is clicked, the GUI executes a tracker with all the desired user's selections.

2.A.4.2. Mote Localization Evaluation and Results

The localization of the sensors was accomplished using the DWMDS provided by the University of Michigan in conjunction with Neal Patwari. The DWMDS is applied onto the Received Signal Strength data from each mote. Basically, each mote transmits a signal in a particular power in 16 different frequencies; every receiving mote records the RSS. Because the signal strength decreases as $1/r^2$ where r is the distance between the transmitting and receiving motes, one can calculate the distance using this propagation model. Once all the data is collected and its dimensionality is reduced using the DWMDS, the locations of all the motes are estimated. The disadvantage of this technique is that it doesn't scale well because each mote keeps a table of all the RSS of every mote and every frequency. Many experiments were conducted by Neal Patwari and Raytheon, and it was found that this technique estimates the location of each sensor with an approximately 0.5 meter MSE. A representative experiment is show in Figure 30 where 16 MICA2 motes were placed at 3 meters apart, and the anchor motes (nodes with known locations) are nodes 1 through node 4.

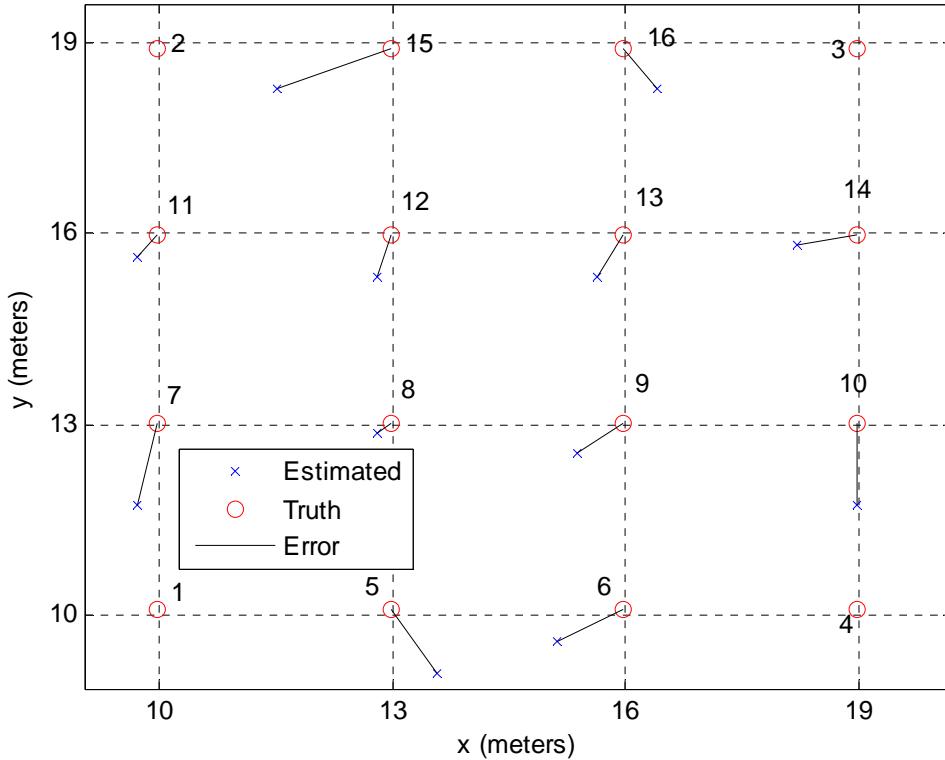


Figure 30: Sensor Localization using DWMDS

2.A.4.3. Tracker Evaluation and Results

In this section, the results of all three trackers are presented with two different targets. The trackers used in our demo are the PF, VM, and UKF, while the targets are the Beeper and the Footstep. The Beeper target is a human walking through the motes field with a Beeper in his hands which emits a beep every three seconds. The Footstep target is also a human walking through the motes field except that the detector on the individual mote is tuned to detect footstep-like sounds (*i.e.*, impulsive sounds of a certain duration).

The Particle Filter Tracker

Particle Filters are usually used to estimate Bayesian models and are analogue to the Markov chain Monte Carlo (MCMC) batch methods. With sufficient particles or samples, they approach the Bayesian optimal estimate. Because they need a large number of particles, they tend to be computationally intensive hence slow. A representative experiment using the PF for a beeping target is shown in Figure 31. In this experiment, 36 motes were used, and they were 10 meters apart in the x and y direction. The color tracks of Figure 31 are as follows. The **magenta** track represents the true track of the target while the **blue** and the **red** track are the PF estimated track. Furthermore, the **blue** track was estimated using all sensors (*i.e.*, all sensors were active), while the **red** track was estimated using fewer sensors. The number of active sensors for the **red** track was determined by the quality of the estimated track during the run. This means that a high quality estimate conserves resources, thus, fewer motes are needed to be active, and a low quality estimate consumes more resources needing more active motes. Therefore, the

objective is to maintain the overall performance of the tracker with fewer resources. This feedback or control mechanism between the tracker and the sensors represents the philosophy of the ISP Program. Figure 31 shows that the **red** track and the **blue** track perform similarly. We would like to point out that the actual sensors of our demo were never turned off (deactivated), but the detections of the sensors marked deactivated were discarded. This is similar to deactivating a sensor without the extra TinyOS code needed.

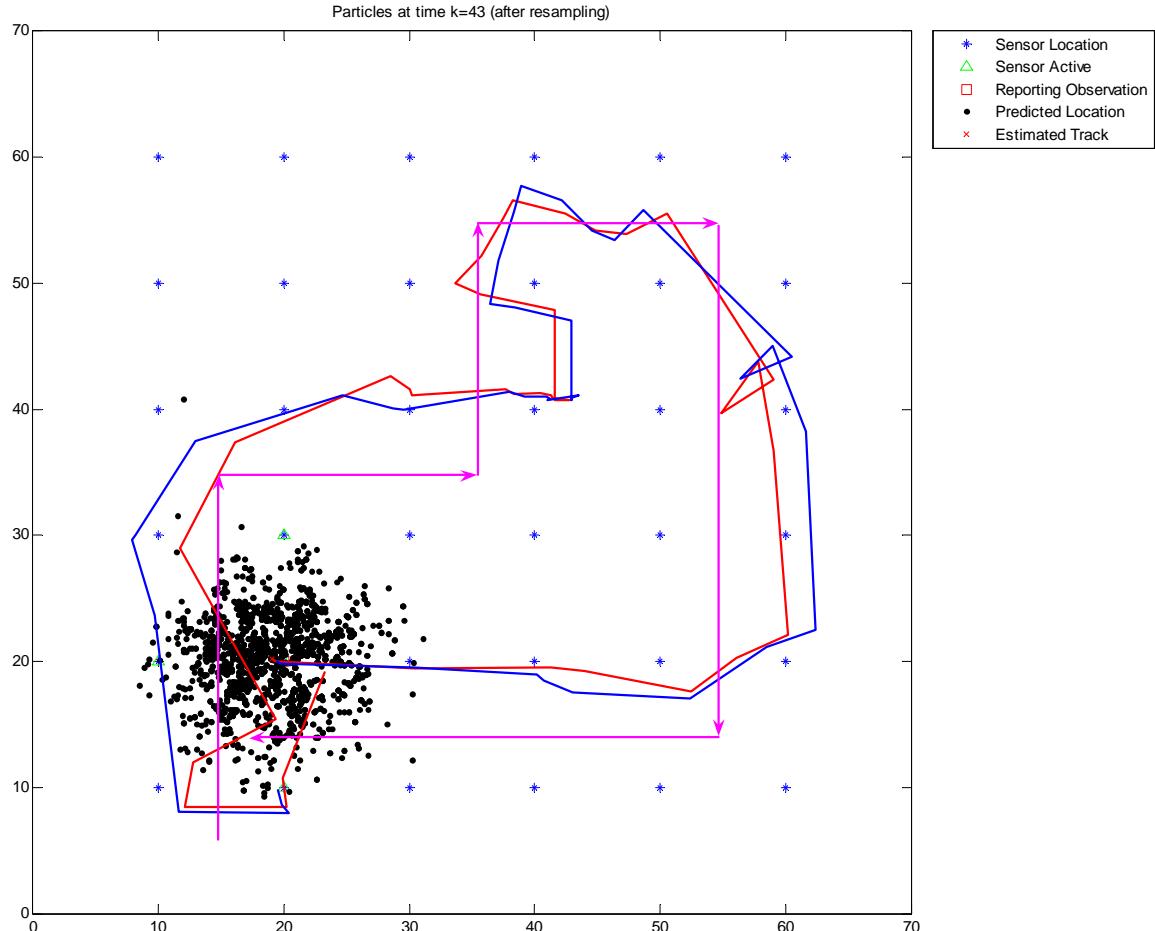


Figure 31: Particle Filter tracker tracking a beeping target

The next experiment is similar to the previous one except that the target is a footstep rather than a beeping target (c.f. Figure 32). The performance of the **red** (with feedback) and **blue** track (without feedback) is similar, once again demonstrating that similar performance can be achieved with fewer resources. However, the estimated tracks differ a lot from the true track (**magenta**). This is due to the footstep detector. This detector doesn't perform as well as the beeper detector due to the higher false alarm rate.

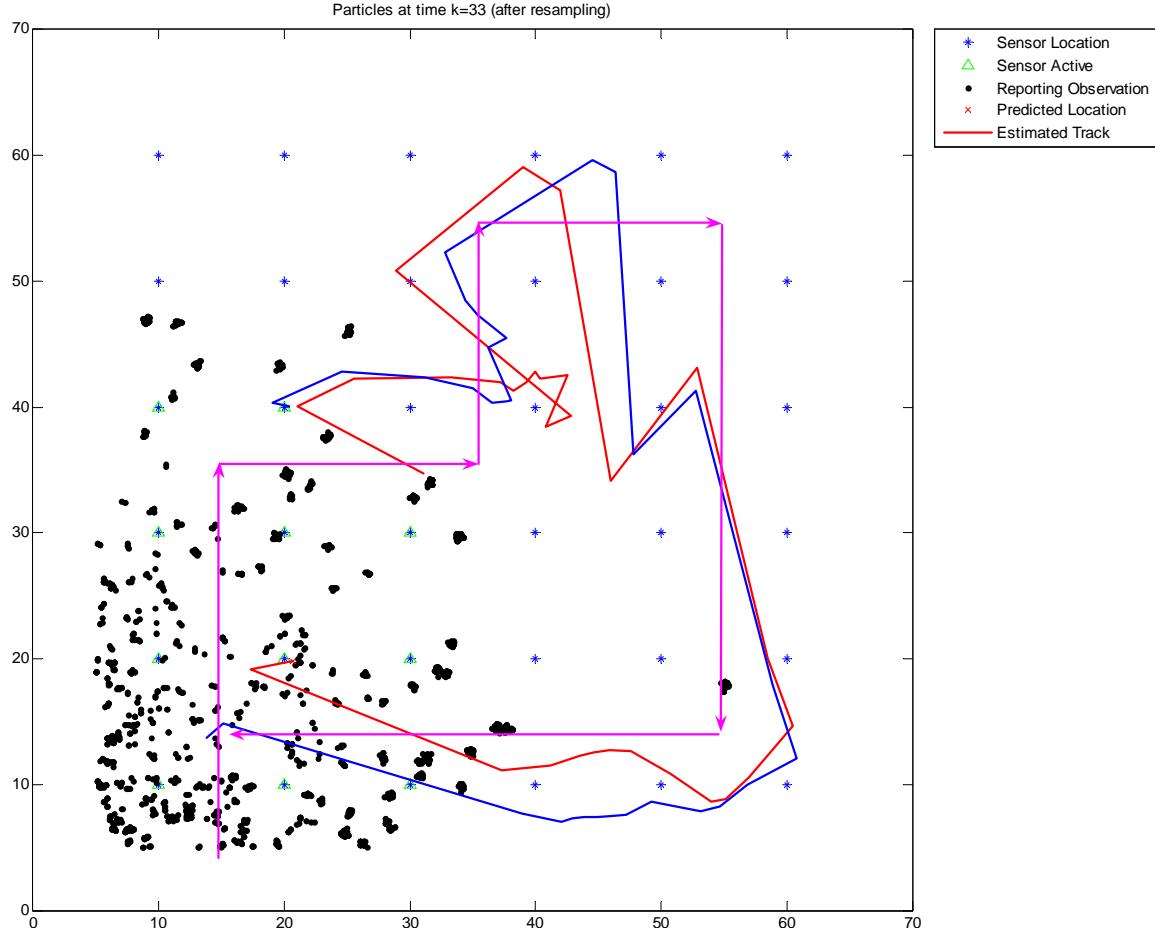


Figure 32: Particle Filter tracker tracking a footstep target.

The Virtual Measurement Tracker

The VM tracker is based on the Kalman Filter. The results are similar to the PF tracker; however, it seems to be more sensitive to false alarms. Once again, 36 motes were used for this experiment, and they were 10 meters apart in the x and y direction. The color tracks of Figure 33 are as follows. The **cyan** track is the estimated track using the VM tracker, and the **magenta** track is the true track of the target. The **cyan** track was computed using all sensors. Due to the lack of time, the feedback mechanism between the tracker and the sensor in order to use fewer resources was not implemented. The VM tracker performed well in tracking the beeping target. On the other hand, the performance of this tracker degraded more than the PF tracker when the footstep detectors were used. The reason is because the footstep detector filters out less false alarms and the VM tracker appears to be more sensitive when many false alarms are present. The performance of the VM tracker in the presence of a footstep target is shown in Figure 34.

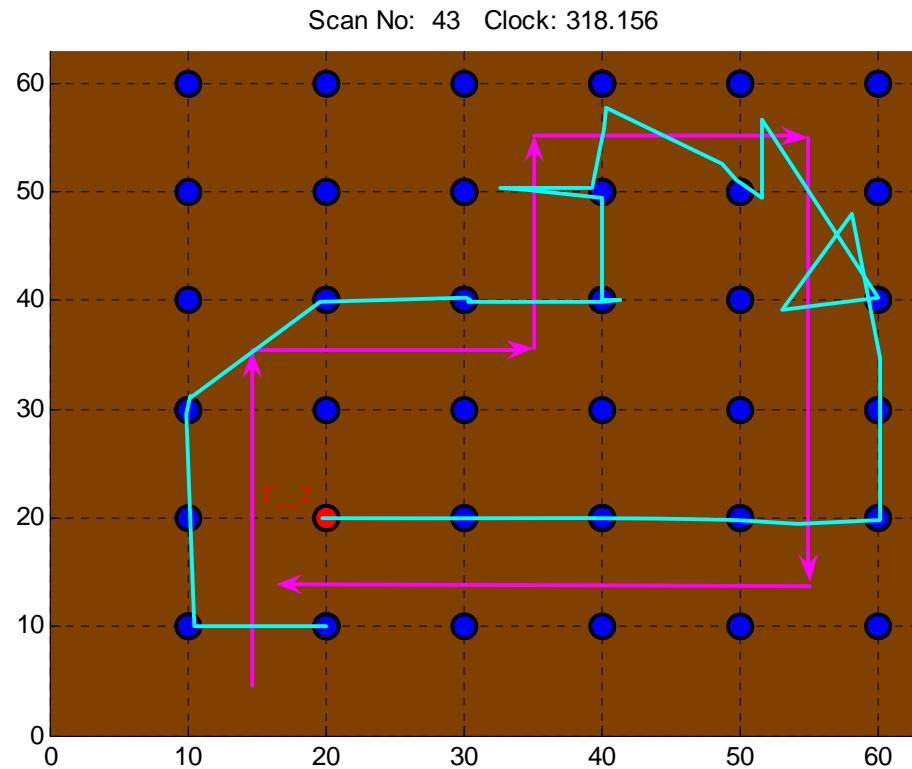


Figure 33: Virtual Measurement tracker tracking a beeping target

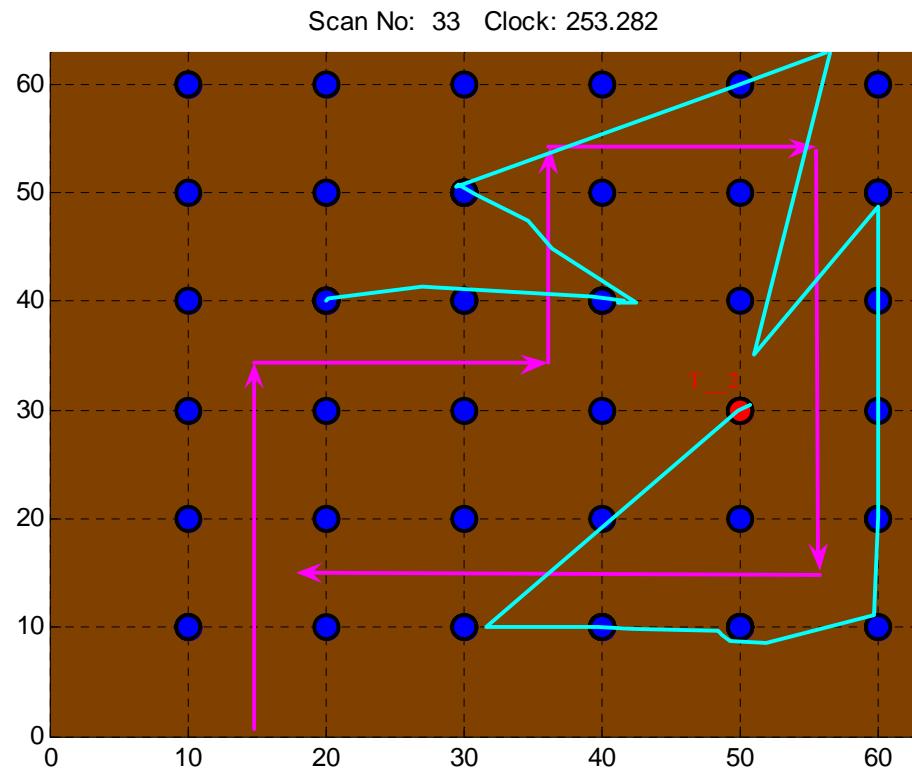


Figure 34: Virtual Measurement tracker tracking a footstep target

The Unscented Kalman Filter Tracker

The UKF tracker is also based on the Kalman Filter. It uses a deterministic sampling technique to pick a minimal set of sample points around the mean. It is a very fast algorithm, and it performs well. Once again, 36 motes were used for this experiment, and they were 10 meters apart in the x and y direction. The color tracks of Figure 35 are as follows. The **green** track is the estimated track using the UKF tracker, and the **magenta** track is the true track of the target. The **green** track was computed using all sensors. Due to the lack of time, the feedback mechanism between the tracker and the sensor in order to use fewer resources was not implemented. The UKF tracker performed well in tracking the beeping target. This tracker didn't perform as well with the footstep target. The reason is because the footstep detector filters out fewer false alarms. The performance of the UKF tracker in the presence of a footstep target is shown in Figure 36.

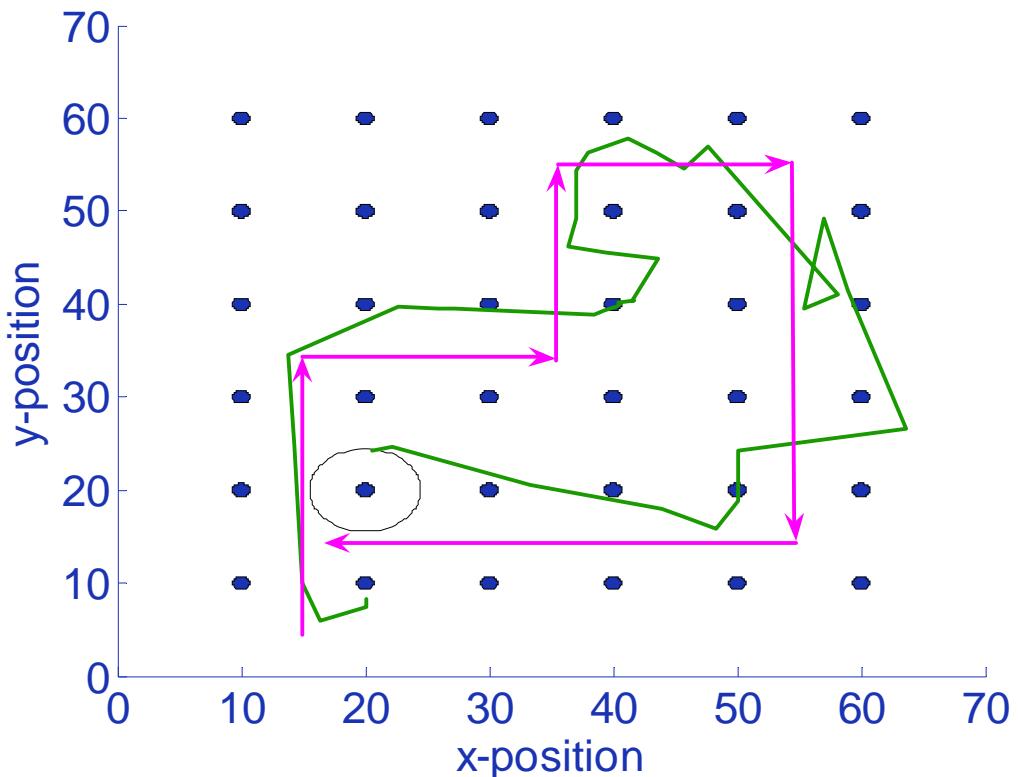


Figure 35: UKF tracker tracking a beeping target

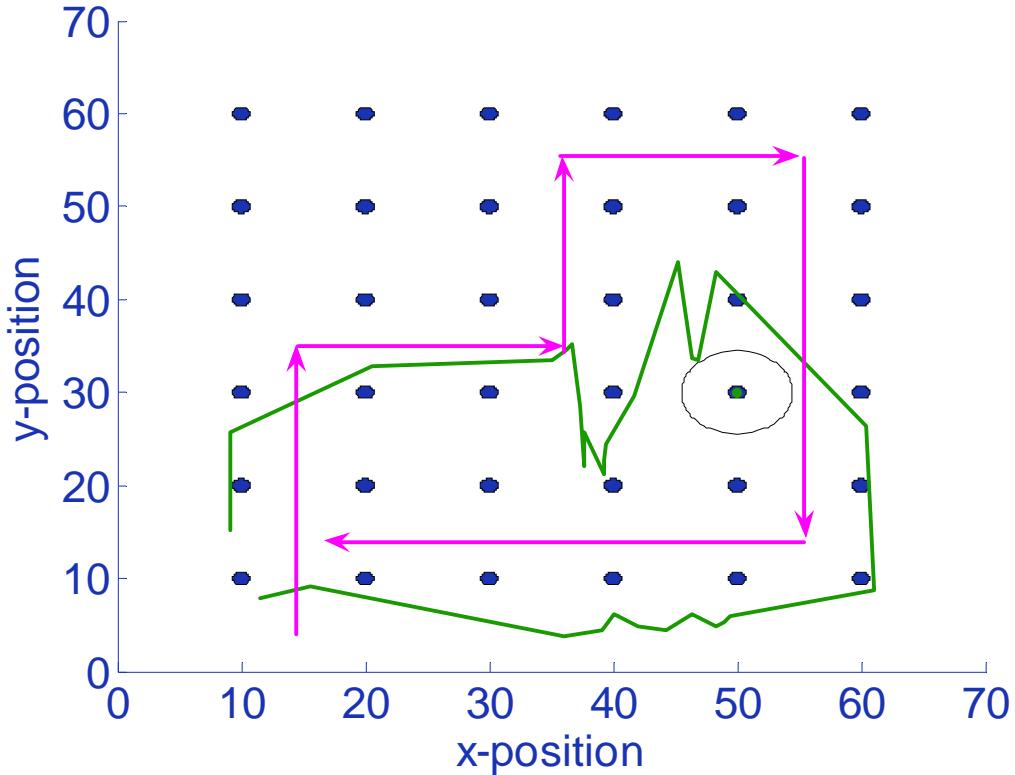


Figure 36: UKF tracker tracking a footstep target

Conclusion

The ISP Multi-Sensor Testbed proved to be a very useful tool in testing some of these ISP algorithms. Our challenge was mostly with the motes as they are difficult to program. However, we were successful in integrating all the parts for the testbed. Unfortunately, due to the lack of time, only the PF tracker had the ability to deactivate sensors. However, it was enough to show that performance didn't degrade with fewer active sensors, and resources were conserved. This was the objective of the ISP Program.

2.4.5. Scheduling of Multiple UAV Platforms for Passive Geolocation

2.A.5.1 Introduction

The late Professor George Dantzig [15] of Stanford “is generally regarded as one of the three founders of linear programming, along with von Neumann and Kantorovich.” As Dantzig reminisced in [16]: “In retrospect, it is interesting to note that the original problem that started my research is still outstanding---namely the problem of planning or scheduling dynamically over time, particularly planning dynamically under uncertainty.” His statement remains an accurate depiction of the difficulty of our scheduling problem, particularly in its most general manifestations.

For this reason, we proceed in a step-by-step fashion, considering first some simpler problems and then gradually increasing the level of difficulty to more realistic, militarily significant problems whose solutions have practical applications. This progression to greater complexity occurs along several paths, *e.g.*, (1) from 2D to 3D spatial geometries, (2) from problems with static targets to constant velocity targets and

ultimately to arbitrarily maneuvering targets, and (3) from single target to multiple target scenarios. As the problems grow in difficulty, the algorithmic solution procedures trend in these directions: from analytic to numerical solutions, from closed form to approximate solutions, and from polynomial time solutions of complexity class P type problems to heuristic solutions of NP-complete type problems.

2.A.5.2 The TDOA Problem

Much previous work has already been done on this problem, *c.f.*, references [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32]. This set of papers covers a wide spectrum of variations in this problem area, ranging from geometric setting (2D or 3D), to solution techniques (*e.g.*, direct, closed form, linearized or iterative), to physical assumptions (*e.g.*, number of emitters, *etc.*), and even the possibility of transforming a TDOA problem (with hyperbolic constraint equations) to a TOA problem (with easier-to-solve spherical constraint equations). This latter possibility occurs when the sensor constellation is numerous enough. Invariably, however, despite their overall high quality, some misconceptions have crept into these works. One such example is the stated number of sensor measurements that are required to compute a unique target location in certain geometric scenarios. This just emphasizes the need for ‘always active’ critical thinking on the part of the reader.

The simplest scenario examined below is the 2D planar problem of using four sensors, with known locations, to determine the location of a single target emitter. Although simple, it is an important building block for more complicated modeling scenarios. Arrival times of an emitted target signal pulse are recorded by the four sensors and differences of these measurements are recorded as TDOAs. These TDOAs form constraint equations and determine hyperbolas in the plane. The site where these hyperbolas intersect marks the target’s location. As can readily be seen from the example scenario portrayed in Figure 37, the usage of only three sensors is, in general, insufficient to uniquely specify a target’s location; hence our basic problem of target position estimation is formulated with four sensors, to remove the possibility of ambiguous solutions.

Our basic “four sensor plus target” analysis scenario includes four sensors located at the corners of a square while the target is presumed to be somewhere in the square’s interior (Figure 38). This latter presumption, however, is not an absolute requirement; useful target locations outside of the square are also very computable, with the caveat of larger error bars. Similarly, the requirement of a square pattern of sensor locations may be significantly relaxed; the software handles general coordinate inputs. Time of Arrival (TOA) measurements are recorded at each sensor; from these four timing data points, a variety of TDOA measurements associated with pairs of sensors may be calculated by simple subtraction. In passive geolocation problems, it is the latter (more indirect) quantities that one must work with to derive the estimated target location.

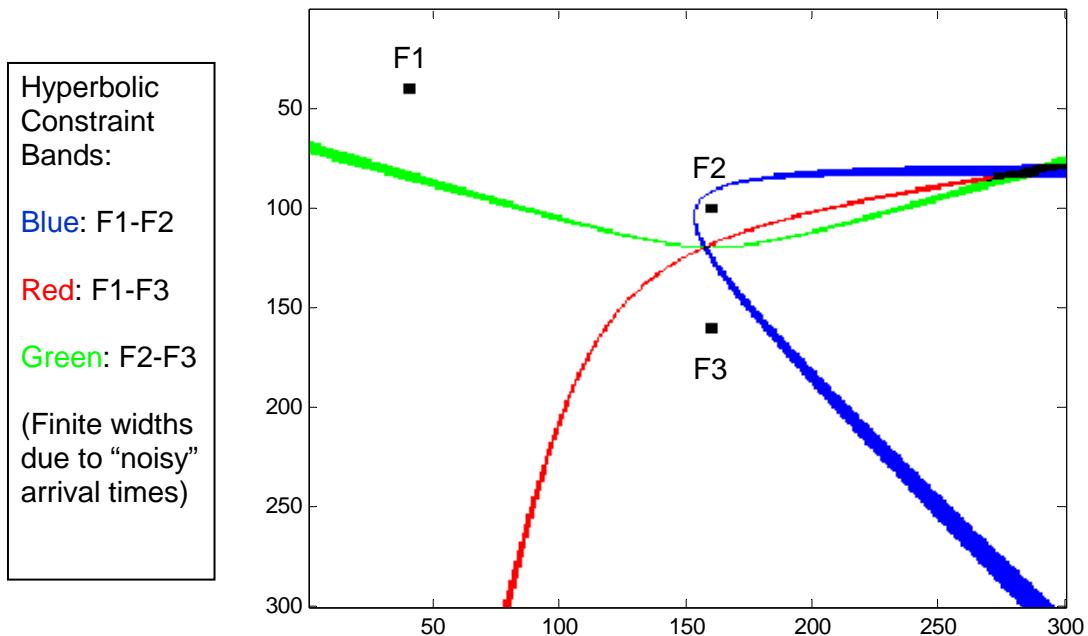


Figure 37: Target location ambiguities arise when using only three sensors (the black squares labeled by: F1, F2 and F3). Three colored hyperbolic envelopes are shown, corresponding to the three pairs of focal points. Two solution regions are shown; these intersection regions are located near the points: (x~160,y~120) and (x~280,y~80)

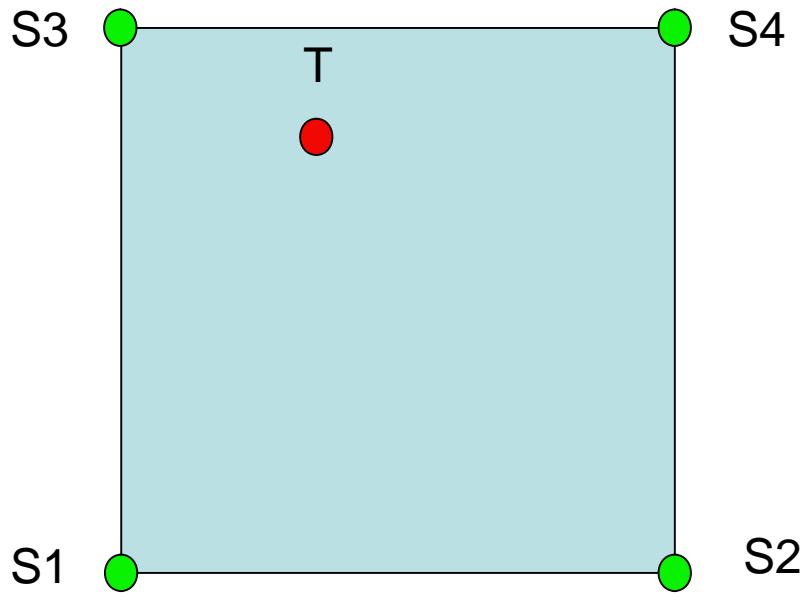


Figure 38: Four sensors (S1 through S4) plus a target (T) in a square geometry

Figure 39 shows the choices made in a commonly used approach for solving the hyperbolic constraint equations. In this approach, a common baseline time point is chosen (in this case, the measurement time at sensor S1) and time differences are only calculated

between this ‘base’ time and the corresponding time measurements obtained at all of the other sensors: S2, S3 and S4.

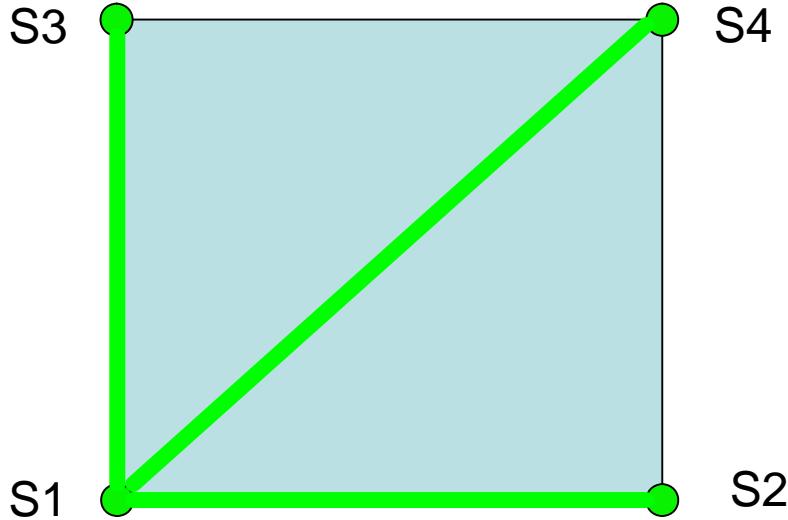


Figure 39: Green lines indicate three TDOA hyperbolic constraint pairs

Figure 40 shows the choices made in our better approach to solving the hyperbolic constraint equations. Here, all possible time differences are calculated (between all possible pairs selected from the four sensors). There are six such pairs, portrayed as three red and three blue lines in the figure.

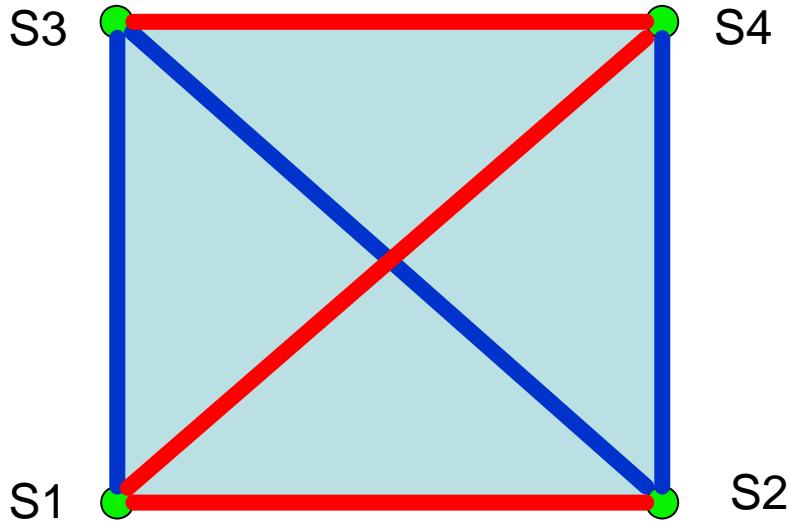


Figure 40: Red and blue lines (three each) indicate a total of six TDOA hyperbolic constraint pairs.

In our analysis several novel features are involved, so a discussion with relevant details is provided here. It starts with a generic description of the hyperbolic constraint equation.

The hyperbolic constraint equation is completely described by the following quantities:

Sensor (focal) points $F1$ and $F2$ with coordinates (x_1, y_1) and (x_2, y_2) , together with arrival times t_1 and t_2 , respectively. The target point T has coordinates (x, y) . The hyperbolic constraint equation is then given by:

$$dist(F1, T) - dist(F2, T) = c(t_1 - t_2).$$

The locus of points T satisfying this equation lies along one branch of a hyperbola. The target lies somewhere along this curve. To obtain a closed-form solution to a pair of such hyperbolic constraint equations, the following manipulations are performed individually, on each of the distinct hyperbolic constraint equations. (Auxiliary parameters are introduced as needed, by context.)

Rewriting the above hyperbolic constraint equation, one has:

$$\sqrt{(x_1 - x)^2 + (y_1 - y)^2} - \sqrt{(x_2 - x)^2 + (y_2 - y)^2} = ct_1 - ct_2$$

$$\sqrt{(x_1 - x)^2 + (y_1 - y)^2} - \sqrt{(x_2 - x)^2 + (y_2 - y)^2} = d_1 - d_2$$

$$\sqrt{(x_1 - x)^2 + (y_1 - y)^2} - \sqrt{(x_2 - x)^2 + (y_2 - y)^2} = d_{12}$$

$$\sqrt{(x_1 - x)^2 + (y_1 - y)^2} = \sqrt{(x_2 - x)^2 + (y_2 - y)^2} + d_{12}.$$

After squaring both sides of the above equation, one obtains:

$$\begin{aligned} (x_1 - x)^2 + (y_1 - y)^2 &= (x_2 - x)^2 + (y_2 - y)^2 + 2d_{12}\sqrt{(x_2 - x)^2 + (y_2 - y)^2} + d_{12}^2 \\ x_1^2 - 2x_1x + y_1^2 - 2y_1y &= x_2^2 - 2x_2x + y_2^2 - 2y_2y + 2d_{12}\sqrt{(x_2 - x)^2 + (y_2 - y)^2} + d_{12}^2 \\ (x_2 - x_1)x + (y_2 - y_1)y + (x_1^2 + y_1^2 - x_2^2 - y_2^2 - d_{12}^2)/2 &= d_{12}\sqrt{(x_2 - x)^2 + (y_2 - y)^2}. \end{aligned}$$

After defining new auxiliary parameters:

$$a_{12}x + b_{12}y + c_{12} = d_{12}\sqrt{(x_2 - x)^2 + (y_2 - y)^2}.$$

Squaring both sides of the equation again yields:

$$\begin{aligned} a_{12}^2x^2 + 2a_{12}b_{12}xy + b_{12}^2y^2 + 2a_{12}c_{12}x + 2b_{12}c_{12}y + c_{12}^2 &= \\ d_{12}^2(x_2^2 - 2x_2x + x^2 + y_2^2 - 2y_2y + y^2) \end{aligned}$$

Collecting like terms, one finally obtains this second degree polynomial in x and y :

$$A_{12}x^2 + B_{12}xy + C_{12}y^2 + D_{12}x + E_{12}y + F_{12} = 0,$$

where the upper case parameters are defined by:

$$\begin{aligned} A_{12} &= a_{12}^2 - d_{12}^2 \\ B_{12} &= 2a_{12}b_{12} \\ C_{12} &= b_{12}^2 - d_{12}^2 \\ D_{12} &= 2(a_{12}c_{12} + d_{12}^2 x_2) \\ E_{12} &= 2(b_{12}c_{12} + d_{12}^2 y_2) \\ F_{12} &= c_{12}^2 - d_{12}^2(x_2^2 + y_2^2) \end{aligned}$$

Thus, each hyperbolic constraint equation is ultimately reducible to the following form:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0.$$

With two such equations, obtained from two different constraint hyperbolas, linear combinations of them may be taken so as to remove either the x^2 term or the y^2 term. Then, after appropriate back substitutions into the original equations, two quartic (*i.e.*, 4th degree) polynomial equations in x alone and in y alone are obtained. These equations are solvable in closed form. Because multiple squarings of equations have occurred in this algorithmic procedure, however, extraneous false solutions (roots) may have been introduced; these must be eliminated.

The false root removal procedure is a simple one: eliminate all but the best solution. The sixteen pairs of (x,y) target location solutions (the Cartesian product derived from all four roots of the quartic equation in x , in combination with all four roots of the quartic equation in y) are substituted in all six of the hyperbolic constraint equations:

$$dist(F1, T) - dist(F2, T) = c(t_1 - t_2)$$

Then, a combined figure of merit Q_k is computed for each of the 16 potential target locations T_k :

$$Q_{k=1\dots 16} = \sum_{m=1}^{6 \text{ or } 3} [dist(F_{1m}, T_k) - dist(F_{2m}, T_k) - c \cdot \Delta t_{1m,2m}]^2$$

The best estimate for the target location is the one that best fits all six (or only three) of the original TDOA equations (in a least squares sense).

Find the k value where the figure of merit Q_k is smallest. Then the best estimate for the target location is given by the (x,y) coordinates corresponding to T_k .

How are the second degree polynomial equations in x and y narrowed down to only two? These are the polynomial equations of the form:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0.$$

One way uses all six of the hyperbolic constraint equations.

From Figure 37, the following three “red” hyperbola equations are added together:

$$A_{12}x^2 + B_{12}xy + C_{12}y^2 + D_{12}x + E_{12}y + F_{12} = 0$$

$$A_{14}x^2 + B_{14}xy + C_{14}y^2 + D_{14}x + E_{14}y + F_{14} = 0$$

$$A_{34}x^2 + B_{34}xy + C_{34}y^2 + D_{34}x + E_{34}y + F_{34} = 0,$$

and the following three “blue” hyperbola equations are added together:

$$A_{13}x^2 + B_{13}xy + C_{13}y^2 + D_{13}x + E_{13}y + F_{13} = 0$$

$$A_{23}x^2 + B_{23}xy + C_{23}y^2 + D_{23}x + E_{23}y + F_{23} = 0$$

$$A_{24}x^2 + B_{24}xy + C_{24}y^2 + D_{24}x + E_{24}y + F_{24} = 0.$$

Note that in Figure 40, the zigzag patterns were chosen so that the combined three “red” hyperbolas and the combined three “blue” hyperbolas would minimize the probability of degenerate situations occurring. (For example, if the four focal points (of two independent hyperbolas) were “parallel”, and the time delays were “just so”, then it is possible that a degeneracy occurs and coincident asymptotic lines could be valid solution sets. This is not good, and would require special handling.)

As a second option (a poorer choice, as will be shown in Figure 41) one may combine only three of the hyperbolic constraint equations.

From Figure 39, the following two hyperbola equations are added together:

$$A_{12}x^2 + B_{12}xy + C_{12}y^2 + D_{12}x + E_{12}y + F_{12} = 0$$

$$A_{14}x^2 + B_{14}xy + C_{14}y^2 + D_{14}x + E_{14}y + F_{14} = 0,$$

and the following two hyperbola equations are added together:

$$A_{13}x^2 + B_{13}xy + C_{13}y^2 + D_{13}x + E_{13}y + F_{13} = 0$$

$$A_{14}x^2 + B_{14}xy + C_{14}y^2 + D_{14}x + E_{14}y + F_{14} = 0.$$

In both of the two optional approaches above, the resulting two summed equations are then processed via the above algorithm to obtain an excellent initial guess for the target location.

Two or three root “polishing” iterative steps are the last ones required to improve the target location estimate. As a byproduct of this analysis procedure, one also obtains information about the target position covariance matrix. The figure of merit parameter, Q , is minimized with respect to changes in the (x,y) coordinates of target location T ; the summation runs over the number of hyperbolas (6 or 3):

$$Q = \sum_{m=1}^{6 \text{ or } 3} \left[\text{dist}(F_{1m}, T) - \text{dist}(F_{2m}, T) - c \cdot \Delta t_{1m,2m} \right]^2$$

Newton-Raphson iterations are used to drive Q to its minimum value (by definition, at this extremum point: $\partial Q/\partial x=0$ and $\partial Q/\partial y=0$). The iterative equation for updating the (x,y) solution points is:

$$\begin{bmatrix} x \\ y \end{bmatrix}_{i+1} = \begin{bmatrix} x \\ y \end{bmatrix}_i - \begin{bmatrix} \partial^2 Q / \partial x^2 & \partial^2 Q / \partial x \partial y \\ \partial^2 Q / \partial x \partial y & \partial^2 Q / \partial y^2 \end{bmatrix}_i^{-1} \cdot \begin{bmatrix} \partial Q / \partial x \\ \partial Q / \partial y \end{bmatrix}_i$$

Although the problem is nonlinear, only two or three iterations are typically needed before machine precision convergence is achieved. This behavior is typical for elliptical paraboloid quadric surfaces, such as $z = Q(x,y)$, especially when the initial solution guess is close to the true solution.

The Hessian matrix is the symmetric 2-by-2 matrix appearing in the above iterative equation. By computing its inverse, one obtains (up to a constant scale factor) the covariance matrix of the measurement errors in the target location coordinates x and y .

Results for this algorithmic approach are given Figure 41. It shows the relative performance of the two ways of computing target locations in this simple scenario.

The four sensors that collect the TOA information are located at the corners of the blue unit square. The ellipses are centered on the ‘decile’ grid points of the target locations---the 121 ‘decile’ points of the unit square, minus the four corner points. The distances of the ellipses from their central target points have all been scaled downward for uncluttered plotting purposes. Note that the blue ellipses always lie within the red ellipses, and they localize the target locations much better than the red ellipses, especially in the lower left corner. Finally, the blue ‘error bars’ obtained from using all 6 TDOA constraints are quite consistent in both size and shape (nearly circular) throughout the interior of the region defined by the four sensors. This is quite useful when processing time-evolving scenarios with the unscented Kalman filter (UKF).

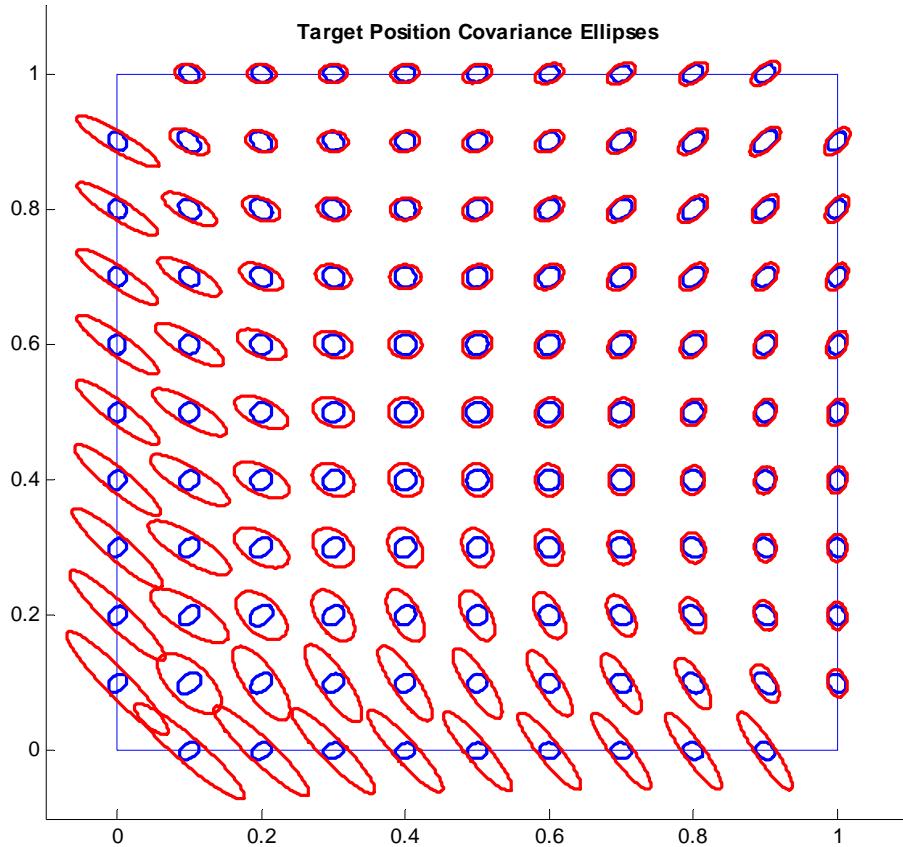


Figure 41: Plot of target position covariance ellipses for the basic scenario, obtained in two ways. The blue ellipses are computed from *all* 6 hyperbolic TDOA constraints, while the red ellipses are computed from *only* 3 such constraints (*i.e.*, TDOAs between sensors located at 0,0 and 1,0; 0,0 and 0,1; and 0,0 and 1,1)

In summary, Figure 41 is a dramatic confirmation of the benefits of fully using the information contained in the four TOA measurements, *i.e.*, fitting *all* 6 hyperbolic equations containing the TDOA information. Reducing the size of the covariance ellipses for estimated target positions is important because the covariance matrix elements are input parameters to the filtering algorithm and, in general, the smaller they are, the less noisy the estimation process for target location will be. An additional side benefit is that, for the same error budget, measurements and analysis will potentially need to be performed less frequently.

2.A.5.3 The Sensor Scheduling Problem

Using computational tools such as the unscented Kalman filter and its various aliases ([33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45]) and approaches such as Partially Observable Markov Decision Processes ([46] [47] [48] [49] [50]), enables analysts to try out novel ideas for solving the sensor scheduling problem of efficiently directing multiple UAVs for passive geolocation ([51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67]). More exotic methods and some survey papers are found in ([68] [69] [70] [71] [72] [73] [74]); these include such topics as coverage in sensor networks via homology concepts.

2.4.6 UniMelb Technical Progress

2.A.6.1 Passive Geolocation Resource Allocation

Introduction

This research investigates resource allocation with geolocation of stationary emitters, using passive sensors located on airborne vehicles (UAVs or manned aircraft, in further text the generic term UAVs is being used). Emitters may be surveillance radars, where the UAVs receive radar pulses either when they are in the main antenna beam, or in the antenna sidelobes; they may also include mobile communication devices.

The chosen method of geolocation is the time difference of arrival (TDOA) of emitter pulses to individual sensors. Possible resources to allocate include: number of UAVs, communication resources, computational resources, choice of received pulses to process (data association), UAV trajectories, *etc.* The resource allocation problems increase dramatically with the number of UAVs, especially in the usual situation of limited communication bandwidth and limited computational resources available onboard the vehicles, against a background of high frequency of pulses emitted. A large number of emitter pulses arrive to each sensor, with a certain probability of detection. The number of possible combinations of received pulses, one per sensor, which need to be investigated to determine a correct combination, grows combinatorially with the number of the sensors. Required communication resource (transmission bandwidth) grows linearly with the number of sensors.

A previous report investigated proof of concept of achieving TDOA emitter geolocation using only two UAVs. Use of two UAVs greatly reduces the communication and computation burden. Instead of all UAVs broadcasting information of all their received pulses to a data fusion center, with considerable computational facilities, only one UAV can broadcast information to surrounding UAVs. Tracking and geolocation can be performed using relatively simple algorithms onboard one of the receiving UAVs. In the work reported, the problem was simplified by assuming no data association problem. We have optimized one resource – number of required UAVs, and also minimized the communication and computational resources necessary.

In this report, we further this approach by assuming Data Association issues, and the resource to be optimized is the choice of received pulses to process. In a common situation, one emitter pulse received on UAV number 1 (UAV#1) can be associated with multiple pulses on UAV#2. Using advanced target tracking techniques, the probability of each received pulse is calculated, and its contribution incorporated in the emitter location estimate. In effect, the pulse probability becomes the cost function. Once that emitter location uncertainty has been reduced, UAV#3 can be chosen to further (dramatically) reduce the emitter location estimation error.

Problem Statement

Emitter location uncertainty when using TDOA measurement from two UAVs, and with one pulse from UAV#2 being associated with four pulses from UAV#1 (*c.f.*, Figure 42). The emitter position uncertainty is shown to be four hyperbolae depicted [75]. When using three UAVs, assume that the central UAV, UAV#2 does the tracking, and that UAV#1 and UAV#3 broadcast their measurements towards UAV#2. Further assume that each received pulse of UAV#2 can be associated with 4 pulses received by

UAV#1 and 4 pulses received by UAV#3, assuming of course that they are able to obtain measurements simultaneously.

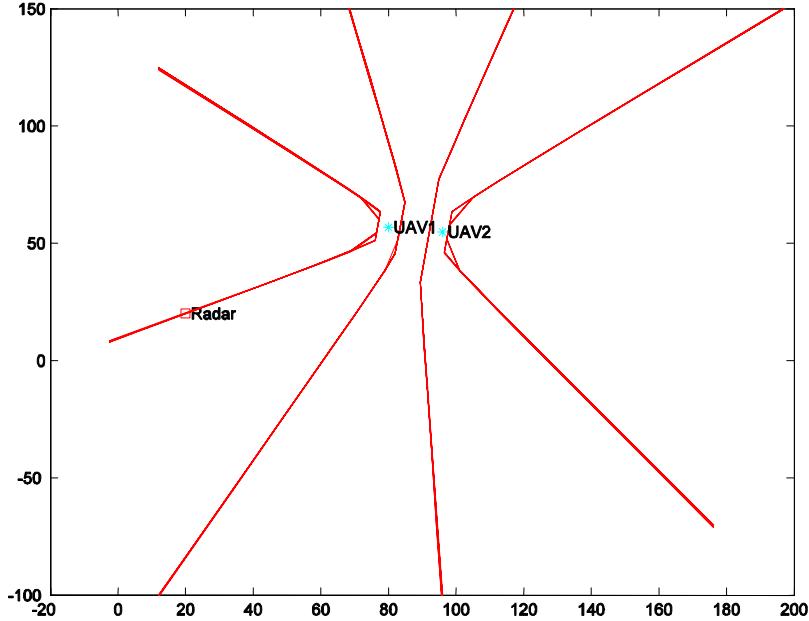


Figure 42: Emitter position uncertainty. Two UAVs, four pulse uncertainty

In that case (*c.f.*, Figure 43) the emitter measurement uncertainty is at the intersections of the four uncertainty hyperbolae obtained using TDOA measurements between UAV#1 and UAV#2 (red hyperbolae) and the four uncertainty hyperbolae obtained using TDOA measurements between UAV#2 and UAV#3 (blue hyperbolae). Even without measurement noise, we have multiple candidates for emitter location (7 in this case), and further UAVs are usually required [76] [77].

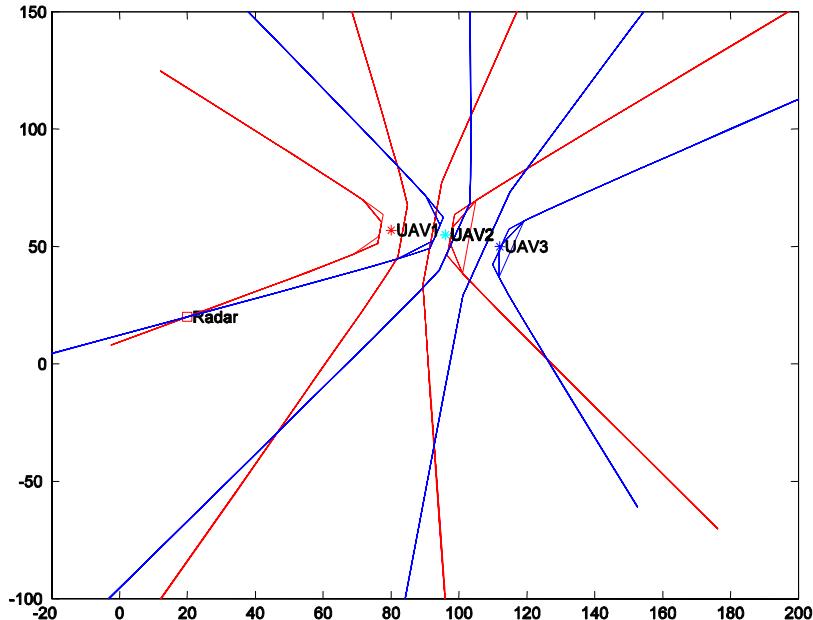


Figure 43: Emitter position uncertainty. Three UAVs, four pulse uncertainty

Solution

To get rid of emitter location uncertainty, we integrate the measurement uncertainty over a number of emitter pulses. A small number of emitter pulses are used by UAV#1 and UAV#2 to reduce emitter location uncertainty to a relatively small region, resolving the Data Association problem. Then the UAV#3 can be chosen to best decrease the emitter location estimation error by using TDOA measurements between UAV#2 and UAV#3.

We first represent the measurement uncertainty, as depicted in Figure 43, by a Gaussian sum [78, 75]. Each hyperbola shown in Figure 43 is not a line, but, due to the time errors or TDOA measurement noise, has width which increases almost linearly with the distance from the center of the two UAVs – Figure 44. This hyperbola area is divided into segments, and each segment is approximated by one Gaussian pdf. Thus the measurement in each scan is a sum of (Gaussian) components.

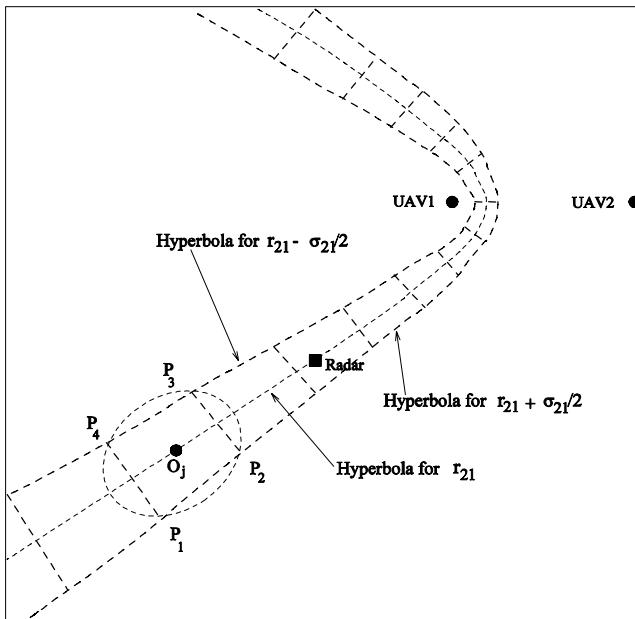


Figure 44: One hyperbola, measurement uncertainty presentation

These measurements are used to initialize and update track, which is an estimate of the emitter location. The track is a simplified version of Integrated Track Splitting (ITS) filter [79, 78, 75], and consists of a number of components. Each track component is a Gaussian pdf which represents the emitter location uncertainty, given a sequence of measurement components. Each track component is defined by Gaussian pdf parameters (mean and covariance), as well as by a relative probability that the component measurement history is correct.

The number of track components grows exponentially in time, and they must be maintained to an acceptable level. A number of sophisticated approaches to track component number reduction have been published [80], however we here use only simple track component pruning, whereas only the components with highest relative probabilities are retained [81].

Numerical Results

We consider the situation depicted in Figure 44, with 3 UAVs. The emitter (radar) is located at coordinates (20km, 20km), and the starting position of the three UAVs are (80,57)km, (96,55)km and (112,50)km. The UAVs travel at the constant speed of 300km/h in the southerly direction. The emitter is a surveillance radar operating in a high PRF mode of 37.5 kHz, which translates to wavefronts 8 km apart. Thus each pulse received by UAV#2 can associate to 4 pulses received by UAV#1 and 4 pulses received by UAV#3. Time measurement errors between two UAVs are assumed to be Gaussian distributed with rms error of 5m, independent from measurement to measurement.

Not every TDOA measurement set is processed; the TDOA measurements which are used are separated 5.5 ms each, to allow for computational delays. Better results will be obtained by integrating all available TDOA measurements. First 5 TDOA measurements are taken from UAV#1 and UAV#2, and Figure 45 depicts the estimated emitter location uncertainty region of the highest probability track components after integrating the 5th TDOA measurement.

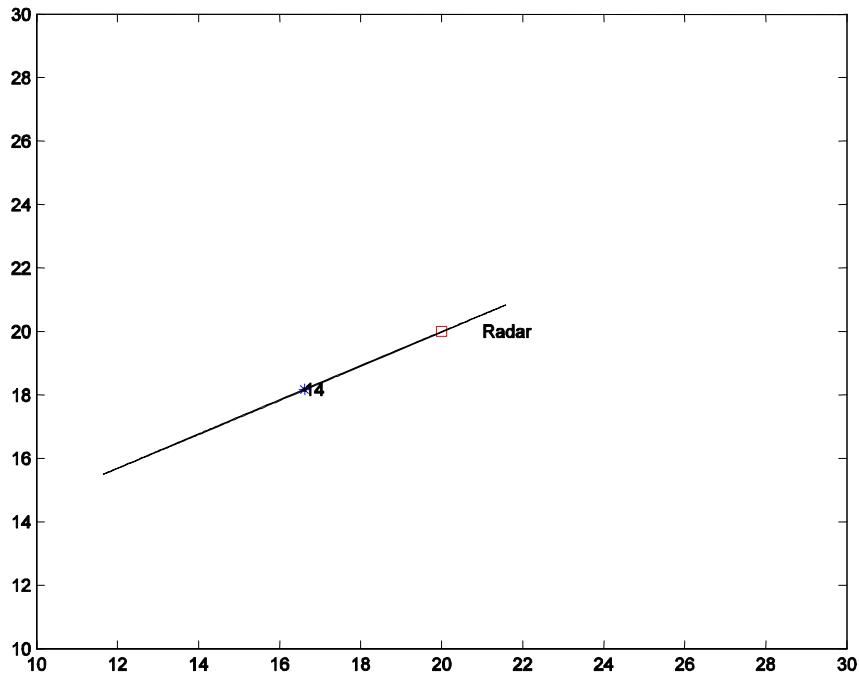


Figure 45: Highest probability track component uncertainty after 5 TDOA measurements. Units are in km.

We assume that at that point TDOA measurements taken from UAV#2 and UAV#3 become available. The ITS tracking filter successfully integrates this additional information, which decreases the estimated emitter location uncertainty region dramatically, as shown in Figure 46. RMS estimation errors are shown in Figure 47, also showing emitter location estimation error decrease at point k=6, when the first TDOA measurement from UAV#2 and UAV#3 is integrated.

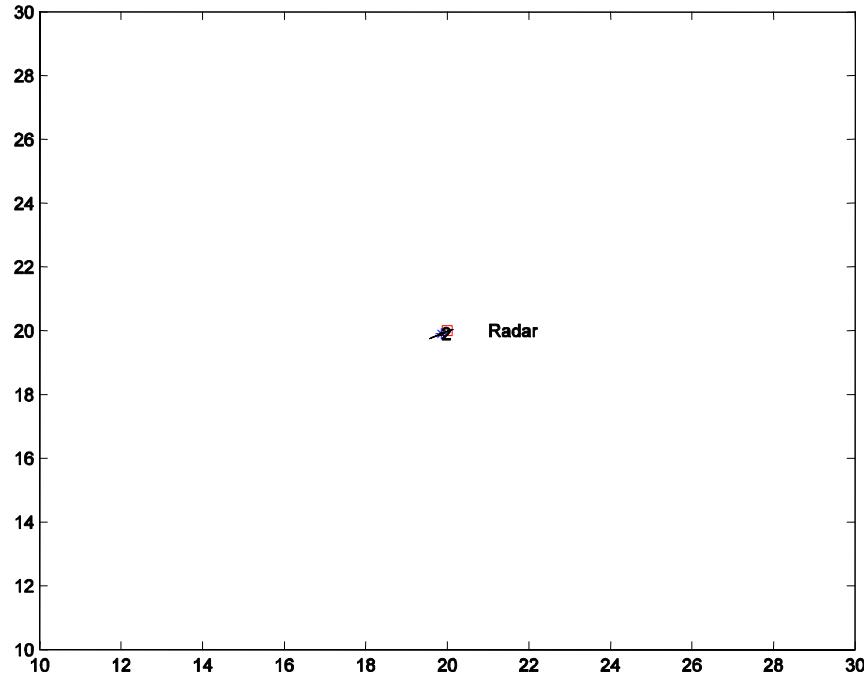


Figure 46: Highest probability track component uncertainty after 6 TDOA measurements.
 Units are in km

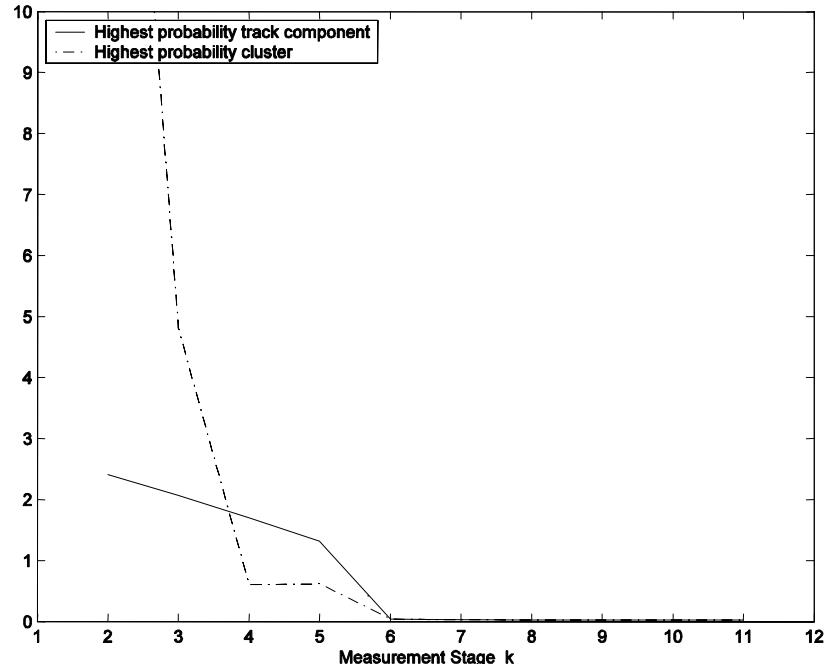


Figure 47: RMS estimation errors over number of TDOA measurements.
 Ordinate units are in km

Conclusions

In this report, passive emitter geolocation using TDOA measurements, with UAV based sensors has been investigated, with a view to minimize and / or optimize resources. Both communication and computation resources have been minimized by using time

measurements from only two UAVs at a time. Data Association (pulse allocation) has been achieved by using Data Association probabilities obtained by sophisticated track updates. This scheme can easily integrate additional time measurements from the third UAV to significantly reduce emitter location estimation errors.

A further work may include uncertain detection of individual pulses by individual sensors (probability of detection) and/or clutter measurements. It is anticipated that using more complete version of the ITS target tracking filter, as described in [78] will take care of both of these issues in a straightforward manner. An effort into establishing the robustness and the probability of track convergence of the proposed scheme is also advised.

2.A.6.2 Unscented Kalman Filter

The results of applying the UKF detection and tracking algorithm to real data are shown in Figure 48. The true target trajectory is indicated by the red solid line and the green dotted line is the filter's estimate of the position of confirmed detected targets. It can be seen that the UKF quickly detects the target and accurately tracks it. No false detections are made.

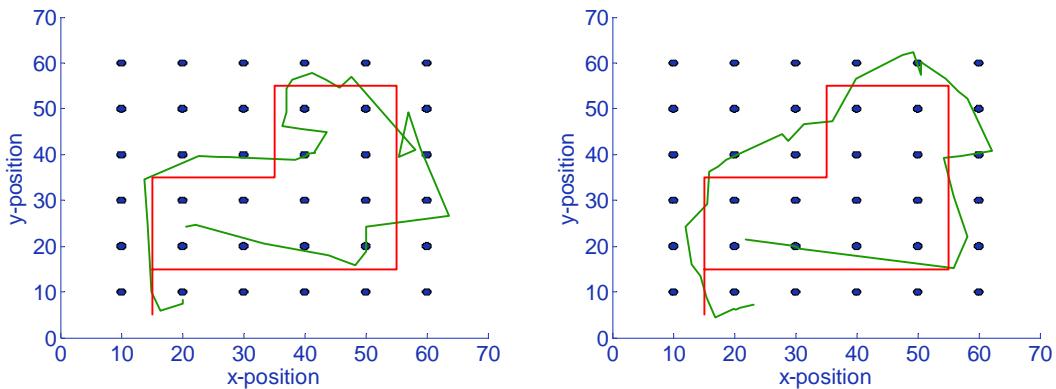


Figure 48: Detection and tracking using the UKF with real data: (a) data file 20070422205553_10m_test1, (b) data file 20070422205553_10m_test2. The red solid line is the true target trajectory. The green dotted line is the estimated trajectory of any confirmed targets.

To obtain the results shown in Figure 48 the returns from all sensors were used in the tracking algorithm. In fact it is not necessary to use all sensor returns. Accurate tracking can be achieved using returns from a subset of returns located about the estimated target position. Let $y_k = [y_{k,1}, \dots, y_{k,m}]$ denote a collection of sensor returns, where m is the total number of sensors, and let $y_{a:b}$ denote the sensor returns collected at times a, \dots, b . Here we select the set

$$D = \{j \in \{1, \dots, m\} : P(y_{k,j} = 1 | y_{t:k-1}) > \Lambda\} \quad (1)$$

where Λ is a pre-defined threshold. The probability of a sensor returning a detection cannot be computed exactly but can be approximated with reasonable accuracy using the unscented transformation. The size of the set D will tend to increase as the uncertainty in the target position increases. Thus the algorithm will adaptively select an appropriate number of motes as the accuracy of the state estimate varies. Figure 49 shows the results

of applying the criterion for $\Lambda = k P_{fa}$, where P_{fa} is the false alarm probability, with $k = 2^5, 2^{10}, 2^{15}$ and 2^{20} . Note that the results of Figure 48 correspond to $\Lambda = P_{fa}$. The average number of sensors used in the various cases are 12.0 for $k=2^5$, 10.0 for $k=2^{10}$, 7.6 for $k=2^{15}$ and 4.1 for $k=2^{20}$. It can be seen that the tracking algorithm responds well to decreases in the number of sensor returns used. There is no discernible difference in performance for values of $k \leq 2^{15}$. For $k=2^{20}$ the algorithm initially loses track of the target but then recovers and tracks with the similar accuracy to that achieved for the smaller values of k .

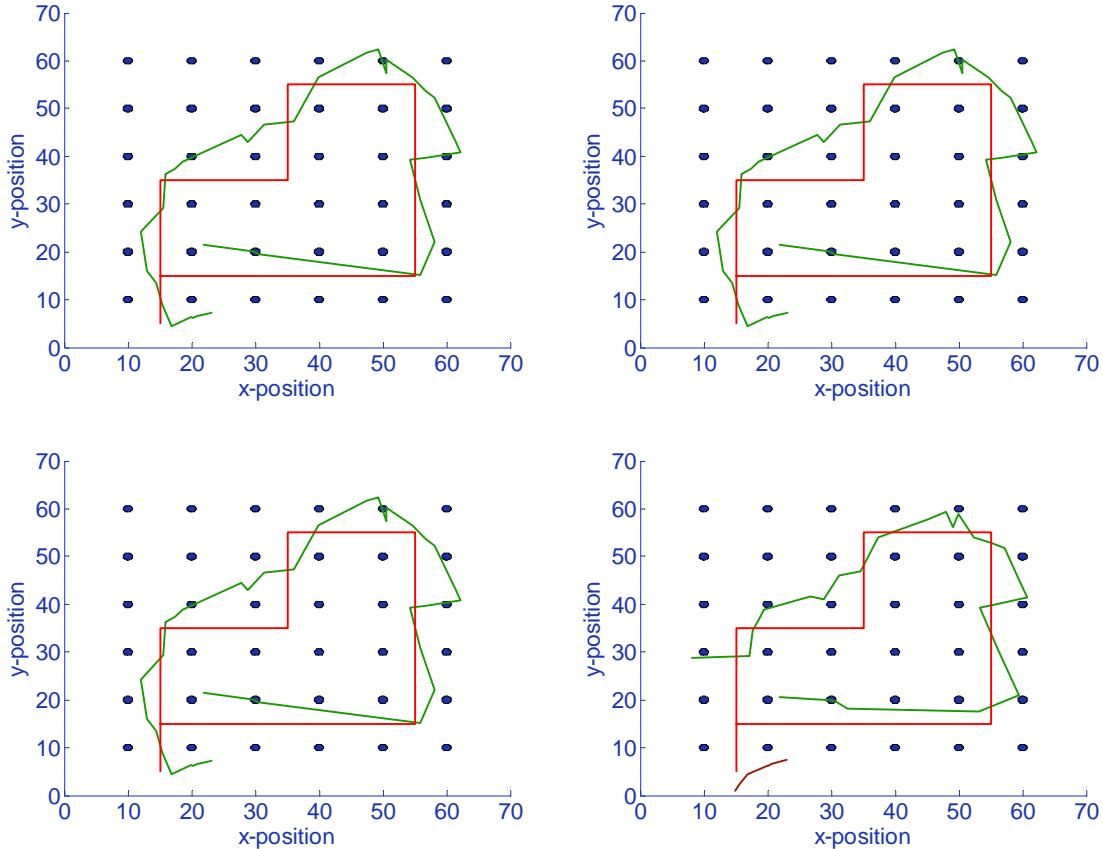


Figure 49: Tracking results with a restricted number of sensors. The criterion of equation (1) is used with $\Lambda = k P_{fa}$ where (a) $k=2^5$, (b) $k=2^{10}$, (c) $k=2^{15}$ and (d) $k=2^{20}$

2.A.6.2 Virtual Measurement Tracker

The Virtual Measurement (VM) multitarget tracker has been integrated into Raytheon tracker testbed software package. The package comes up with some real data obtained by using two types of sensors on board of a mote. By playing with the data available we may see:

- With a high SNR threshold, sensor detection sequence is just a sequence of single activated (including null activated) mote index. In this case, the VM tracker acts just like an activated mote follower of natural choice.

- In the case of low SNR or higher detection sensitivity, multiple motes can be activated from a single scan. However, higher false detection rate can be observed such as those data from the detection of footsteps.
- As no ground truth is available, we cannot make a reasonable performance comparison rather than see how smooth and close the estimated position trajectory is to the true trajectory.

In this report, we compare the tracker performance by using different parameters which can be modified via the set track.m or the input of GUI.

Choice of Sensing Range

The choice of sensing range can be made through the GUI input dialog box. In VM tracker, for a fixed mote distribution and known sensing range, a larger sensing range will result in a lower virtual measurement noise and therefore can achieve a better tracking performance. However, this is definitely not the case with the data at hand, as the actual detection (sensing) range is changing from time to time. We compared the estimated trajectories when using three different sensing ranges (3, 5, 7 meters). Intuitively, when sensing range = 7 m is chosen the trajectory looks more close to the true one on average.

Choice of Process Noise Factor

The underlying target motion model for the data is unknown. The current version of VM tracker assumes a target motion on a constant velocity model with a Gaussian zero-mean noise controlled by the system process noise factor tracker.w1 in set track.m. We repeated the scenario test using different values of tracker.w1. Our test result indicates that a smaller value of tracker.w1 = 0.01 can result in a smoother trajectory.

Choice of TPM for Track Quality Measure

In the VM tracker, there is an underlying Markov chain model for track existence. The track life is monitored from the probability of target existence calculated by the tracker based on the underlying Markov transition probability matrix (TPM). The default value for the TPM is [0:98 0:02; 0 1] and it may be changed by modifying set track.m. Two cases were considered here: Case 1: default and Case 2: [0:8 0:2; 0 1].

Conclusion

From experiment, we found that the VM tracker is sensitive to the choice of sensing range, and process noise factor while it is insensitive to the detection probability choice. A recommend parameter set for this scenario is

sensingrange = 7m and *w1* = 0.01

The VM tracker is a multitarget tracking algorithm which can pick up a dead track or initiate new track by splitting a track due to multiple target measurements. Such an example can be found in playing data file 20070329095827 footstep.pbd, where sensing range = 7 m, *w1* = 0.01.

To obtain a better tracking performance with respect to such unstable (noisy) target movement, we should design a better system model for the VM tracker by using IMM structure.

2.A.7 University of Michigan Technical Progress

The last reporting period ended on 2/19/07 and the ISP subcontract terminated on 3/31/07. Thus this report covers work only a six week period. We have wrapped up our research in classification constrained dimensionality reduction; self localization; and geometric entropy minimization (GEM). Our principal activities were:

- Completion a report on the out-of-sample extension of CCDR that includes simultaneous updates for both unlabeled and labelled data.
- This report is attached. A summary of the findings are included below. Al Hero visited Neal Patwari at the University of Utah in late February 2007. During this visit we discussed the issues that Raytheon was running into with implementation of the dwMDS algorithm for self localization of their wireless sensor test bed. The dwMDS algorithm was designed to work with RSS measurements but, if some sort of time synchronization is feasible, it is also applicable to time delay measurements which may be more reliable for the scenario (larger inter-sensor distances) that Raytheon is using. We believe that the issues of convergence encountered by Raytheon are due to a software bug.

OSE for Classification constrained dimensionality reduction

At the November 2006 ISP Phase II PI meeting we presented results for the out-of-sample extension (OSE) of our classification constrained dimensionality reduction (CCDR) that can only be applied to unlabeled data, *e.g.* test samples to be classified. This period we have concentrated on extension of CCDR to labelled as well as unlabeled data. An outline of the extension is given below.

Let $\{x_1, \dots, x_n\}$ be high-dimensional samples, and $\{y_1, \dots, y_n\}$ be their lower-dimensional (d -dimensional) embedding found by SVD. As usual, define Λ as the $d \times d$ diagonal matrix of the first d eigenvalues of the Graph Laplacian. For a new unlabeled data point x_{n+1} , the out-of-sample-extension for an unlabeled point is

$$y_{n+1}(x_{n+1}) = \Lambda^{-1} \frac{\sum_{i=1}^n K(x_i, x_{n+1}) y_i}{\sum_{s=1}^n K(x_s, x_{n+1})} \quad (1)$$

For a new labelled data point x_{n+1} which belongs to class k we can show the modified out-of-sample-extension

$$y_{n+1} = \Lambda^{-1} \left(\beta \frac{\sum_{i=1}^n K(x_i, x_{n+1}) y_i}{\beta \sum_{s=1}^n K(x_s, x_{n+1}) + 1} + \frac{z_k}{\beta \sum_{s=1}^n K(x_s, x_{n+1}) + 1} \right) \quad (2)$$

where z_k is the centroid for class k , given by the SVD performed for the first n points, and β is the regularization parameter applied to the centroid points. A Matlab program has been provided to Raytheon.

2.A.8 Georgia Tech Technical Progress

2. B. Refereed Publications

There were three refereed publications during the current PoP.

1. S. Sira, A. Papandreou-Suppappola and D. Morrell, ``Dynamic Configuration of Time-varying Waveforms for Agile Sensing and Tracking in Clutter," *IEEE Transactions on Signal Processing*, in print, 2007.
2. A. Chhetri, D. Morrell and A. Papandreou-Suppappola, ``On The Use of Binary Programming for Sensor Scheduling," *IEEE Transactions on Signal Processing*, in print, 2007.
3. A. Chhetri, D. Morrell and A. Papandreou-Suppappola, ``Non-myopic Sensor Scheduling and its Efficient Implementation for Target Tracking Applications," *EURASIP Journal on Applied Signal Processing*, vol. 2006, Article ID 31520, 18 pages, 2006.
4. S. Chhetri, D. Morrell and A. Papandreou-Suppappola, ``Sensor Resource Allocation for Tracking Using Outer Approximation", *IEEE Signal Processing Letters*, vol. 14, no. 3, pp. 213-216, 2007.
5. I. Kyriakides, D. Morrell and A. Papandreou-Suppappola, ``Sequential Monte Carlo Methods for Tracking Multiple Targets with Deterministic and Stochastic Constraints," *IEEE Transactions on Signal Processing*, under second revision, May 2007.

2. C. Conference Proceedings

There were four publications in conference proceedings during the current PoP.

1. ``Sparse Manifold Learning with Applications to SAR Image Classification," V. Berisha, N. Shah, D. Waagen, H. Schmitt, S. Bellofiore, A. Spanias, and D. Cochran, 32nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Honolulu, HI, April 15-20, 2007.
2. N. Okello and D. Musicki, ``Measurement Association for Emitter Geolocation with Two UAVs," 10th International Conference on Information Fusion, Québec City, Canada, 9-12 July 2007, accepted.
3. X. Wang and D. Musicki, ``Target Tracking Using Energy Based Detections," 10th International Conference on Information Fusion, Québec City, Canada, 9-12 July 2007, accepted.
4. D. Musicki, ``Target Existence Based Resource Allocation," 10th International Conference on Information Fusion, Québec City, Canada, 9-12 July 2007, accepted.

2. D. Consultative and Advisor Functions

There were no consultative or advisory functions that occurred during the current PoP. However, over the course of the Raytheon ISP Phase II program, we have developed a significant relationship with Raytheon Shooter Localization demonstration using the MICA-2/Z sensor nodes. This work is being funded under the DARPA IXO NEST Phase II program. The Phase I shooter localization algorithms were developed by VU. Preliminary results indicated that the shooter localization algorithm has significant potential. The program was subsequently classified and was ultimately transitioned to Raytheon for demonstration and refinement under Phase II. The DARPA IXO Program

ISP Phase II (Contract N00014-04-C-0437)
Final Technical Progress Report (CDRL A004 No. 1)

Manager has given permission for several of these algorithms to be used in our program. The Raytheon NEST program has identified a critical need for the development of an accurate sensor localization algorithm that is scalable to hundreds or thousands of nodes. We have kept the Raytheon NEST program informed as to the progress of our sensor localization research.

2. E. New Discoveries, Inventions or Patent Disclosures

There were no patent disclosures filed during the current PoP nor, unfortunately, over the course of the program.

2. F. Honors/Awards

There were no honors or awards received during the current PoP nor, unfortunately, over the course of the program.

2. G. Transitions.

There were no specific technology transitions achieved during the current PoP. Raytheon ISP Phase II personnel also support the AFRL Small Diameter Bomb (SDB) program and have managed to transition ISP methodology. Finally, the Raytheon ISP Phase II program has transitioned a number of manifold extraction ideas into the Raytheon ATR Enterprise Campaign.

2. I. Acronyms

ADTS	Advanced Detection Technology Sensor
ASU	Arizona State University
ATA	Automatic Target Acquisition
AVU	Algorithms Verification Units
CAD	Computer-Aided-Design
CADSP	Cooperative Analog Digital Signal Processor
CCCD	Class Cover Catch Diagraphs
CCDR	Classification Constrained Dimensionality Reduction
CPI	Coherent Processing Interval
CRB	Cramér–Rao Bound
CROPS	Classification Reduction Optimal Policy Search
DARPA	Defense Advanced Research Projects Agency
DS	Danzig Selector
DSA	Distinct Sensing Area
dwMDS	Distributed, weighted, multi-dimensional scaling
FPA	Focal Plane Array
FMAH	Fast Mathematical Algorithms and Hardware
GEM	Geometric Entropy Maps
Georgia Tech	Georgia Institute of Technology
GMS	Gauss-Markov Systems
GPS	Global Positioning System
IASG	Independently Activated Sensor Group
ISP	Integrated Sensing and Processing
IXO	Information Exploitation Office
kNN	k-Nearest Neighbor
LEAN	Laplacian Eigenmap Adaptive Neighbor
LIP	Linear Integer Programming

M2M	Multipoint-to-multipoint
MC	Monte-Carlo
MTT	Multi-target tracking
NEST	Networked Embedded System Technology
NDA	Non-disclosure Agreement
NLIP	Nonlinear Integer Programming
NLOS	NetFires Non-Line of Sight
NUC	Non-Uniformity Compensation
ONR	Office of Naval Research
OSE	Out-of-sample extension
PAM	Precision Attack Munition
PDA	Probabilistic Data Association
PRI	Pulse Repetition Intervals
PWF	Polarization Whitening Filter
PoP	Period of Performance
RIM	Radio Interferometric Measurements
RIPS	Radio Interferometric Positioning
RISCO	Raytheon International Support Company
RSS	Received Signal Strength
TAA	Technical Assistance Agreement
TDOA	Time Difference of Arrival
TIM	Technical Interchange Meeting
TID	Threat Identification
UAV	Unmanned Aerial Vehicle
UCIR	Uncooled infrared imaging
UKF	Unscented Kalman filter
UM	University of Michigan
UniMelb	Melbourne University
VM	Virtual Measurement
VU	Vanderbilt University

2. J References

- 1 <http://www.xbow.com/>
- 2 M. Maroti, B.Kusy, G. Balogh, P. Volgyesi, K. Molnar, A. Nadas, S. Dora and A. Ledeczi, "Radio Interferometric Geolocation," Vanderbilt University, Technical Report ISIS-05-602, 2005.
- 3 J. G. DeVinney, C. E. Priebe, D. J. Marchette, and D. A. Socolinsky, "Random Walks and Catch Digraphs in Classification," *preprint*, 2002.
- 4 C. E. Priebe, D. J. Marchette, J. G. DeVinney, and D. A. Socolinsky, "Classification Using Class Cover Catch Digraphs," *Journal of Classification*, **20**, pp. 3-23, 2003.
- 5 A. Johnson, "Spin Images: A Representation for 3-D Surface Matching", Doctoral dissertation, *Tech. Report CMU-RI-TR-97-47, Robotics Institute, Carnegie Mellon University*, 1997.

6 A. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**, pp. 433-449, 1999.

7 N. Henze and M. Penrose, "On the Multivariate Runs Test," *Annals of Statistics*, **27**, pp. 290-298, 1999.

8 J. Friedman and L. Rafsky, "Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests," *Annals of Statistics*, Vol. 7, pp. 697-717, 1979.

9 T. K. Ho and M. Basu, "Complexity Measures of Supervised Classification Problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, pp. 289-300, 2002.

10 M. Á. Carreira-Perpiñán, "A Review of Dimension Reduction Techniques," *University of Sheffield, Department of Computer Science Technical Report*, **CS-96-09**, pp. 1-69 (1997).

11 J. L. Solka. and D. A. Johannsen, "Classifier Optimization Via Graph Complexity Measures," in B. A. Bodt and E. J. Wegman (eds.) *Proceedings of the Eighth Annual U.S. Army Conference on Applied Statistics*, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, pp. 121-135 (2003).

12 F. Lebourgeois and H. Emptoz, "Pretopological Approach for Supervised Learning," *Proceedings of the 13th International Conference on Pattern Recognition*, pp. 256-260 (1996).

13 Skolnick, *Introduction to Radar Systems*, McGraw-Hill, 2001.

14 H. Van Trees, Optimum Array Processing, John Wiley & Sons, New York, 2002.

15 R. Freund, "Professor George Dantzig: Linear Programming Founder Turns 80," *SIAM News*, November 1994; <http://www.stanford.edu/group/SOL/dantzig.html>.

16 G.B. Dantzig, chapter on Linear programming, in *History of Mathematical Programming: A Collection of Personal Reminiscences*, J.K. Lenstra, A.H.G. Rinnooy Kan and A. Schrijver (eds.), Elsevier, 1991.

17 D.J. Torrieri, "Statistical Theory of Passive Location Systems," *IEEE Trans. Aerospace Electronic Systems*, vol. 20, no. 2, March 1984, p. 183-198.

18 K.C. Ho and Y.T. Chan, "Solution and Performance Analysis of Geolocation by TDOA," *IEEE Trans. Aerospace and Electronic Systems*, vol. 29, no. 4, Oct. 1993, p. 1311-1322.

19 Y.T. Chan and K.C. Ho, "A Simple and Efficient Estimator for Hyperbolic Location," *IEEE Trans. Signal Processing*, vol. 42, no. 8, Aug. 1994, p. 1905-1915.

20 R. Schmidt, "Least Squares Range Difference Location," *IEEE Trans. Aerospace Electronic Systems*, vol. 32, no. 1, January 1996, p. 234-242.

21 K.C. Ho and Y.T. Chan, "Geolocation of a Known Altitude Object from TDOA and FDOA Measurements," *IEEE Trans. Aerospace Electronic Systems*, vol. 33, no. 3, July 1997, p. 770-783.

22 C. Ma, R. Klukas and G. Lachapelle, "An Enhanced Two-Step Least Squared Approach for TDOA/AOA Wireless Location," *IEEE International Conference on Communications*, 2003, vol. 2, 11-15 May 2003, p. 987-991.

23 G. Mellen, II, M. Pachter and J. Raquet, "Closed-Form Solution for Determining Emitter Location Using Time Difference of Arrival Measurement," *IEEE Trans. Aerospace Electronic Systems*, vol. 39, no. 3, July 2003, p. 1056-1058.

24 S.P. Drake and K. Dogancay, "Geolocation by Time Difference of Arrival Using Hyperbolic Asymptotes," *Acoustics, Speech, and Signal Processing, 2004, Proc., IEEE International Conference on*, 17-21 May 2004, vol. 2, p. II-361 to II-364.

25 K.C. Ho and W. Xu, "An Accurate Algebraic Solution for Moving Source Location Using TDOA and FDOA Measurements," *IEEE Trans. Signal Processing*, vol. 52, no. 9, Sept. 2004, p. 2453-2463.

26 K. Dogancay, "Emitter Localization using Clustering-Based Bearing Association," *IEEE Trans. Aerospace Electronic Systems*, vol. 41, no. 2, April 2005, p. 525-536.

27 W.-C. Li, P. Wei and X.-C. Xiao, "TDOA and T2/R radar based target location method and performance analysis," *IEE Proc.-Radar, Sonar and Navigation*, vol. 152, no. 3, June 2005, p. 319-323.

28 C.O. Savage, H.A. Schmitt, R. Cramer and Bill Moran, "Dynamic Positioning and Scheduling UAVs for Passive Geolocation," November 30, 2005.

29 T. Sathyam, A. Sinha and T. Kirubarajan, "Passive Geolocation and Tracking of an Unknown Number of Emitters," *IEEE Trans. Aerospace Electronic Systems*, vol. 42, no. 2, April 2006, p. 740-750.

30 C.O. Savage, R.L. Cramer, H.A. Schmitt, "TDOA Geolocation with the Unscented Kalman Filter".

31 J. Nie, "Sum of Squares Method for Sensor Network Localization,"
[http://arxiv.org/math/0605652 \(2006\)](http://arxiv.org/math/0605652).

32 W.-K. Ma, B.-N. Vo, S. Singh and A. Baddeley, "Tracking an Unknown Time-Varying Number of Speakers using TDOA Measurements: A Random Finite Set Approach," *IEEE Trans. Signal Processing*, vol. 54, no. 9, Sept. 2006, p. 3291-3304.

33 R. van der Merwe, A. Doucet, N. De Freitas and E. Wan, "The Unscented Particle Filter," Cambridge University Engineering Dept., Technical Report CUED/F-INFENG/TR 380, 2000.

34 S.J. Julier and J.K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proc. IEEE*, vol. 92, no. 3, March 2004, p. 401-422.

35 G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *J. Geophysical Research*, vol. 99, no. C5, p. 10143-10162, May 15, 1994.

36 G. Evensen, "The Ensemble Kalman Filter: theoretical formulation and practical implementation," *Ocean Dynamics* (2003) vol. 53, p. 343-367.

37 G. Evensen, "Sampling strategies and square root analysis schemes for the EnKF," *Ocean Dynamics* (2004) vol. 54, p. 539-560.

38 D. Mackenzie, "Ensemble Kalman Filters Bring Weather Models Up to Date," from *SIAM News*, vol. 36, no. 8, October 2003.

39 B.R. Hunt, E.J. Kostelich and I. Szunyogh, "Efficient Data Assimilation for Spatiotemporal Chaos: a Local Ensemble Transform Kalman Filter,"
[http://arxiv.org/physics/0511236 \(2005\)](http://arxiv.org/physics/0511236).

40 J.H. Gove and D.Y. Hollinger, "Application of a dual unscented Kalman filter for simultaneous state and parameter estimation in problems of surface-atmosphere exchange," *J. Geophysical Research*, vol. 111, D08S07, doi:10.1029/2005JD006021, 2006.

41 G.-J. Zhang, Y. Yao and K.-M. Ma, "Line of sight rate estimation of strapdown imaging guidance system based on unscented Kalman filter," *Proc. of 2005 International Conference on Machine Learning and Cybernetics*, 18-21 August 2005, vol. 3, p. 1574-1578.

42 S. Gillijns, O.B. Mendoza, J. Chandrasekar, B.L.R. De Moor, D.S. Bernstein and A. Ridley, "What is the Ensemble Kalman Filter and How Well Does It Work?," *Proc. of the 2006 American Control Conference*, 14-16 June 2006, doi:10.1109/ACC.2006.1657419.

43 I. Hoteit, G. Triantafyllou and G. Korres, "Using Low-Rank Ensemble Kalman Filters for Data Assimilation with High Dimensional Imperfect Models," *J. of Numerical Analysis, Industrial and Applied Mathematics*, vol. 2, no. 1-2, 2007, p. 67-78.

44 R.T. Azuma, "Correcting for Dynamic Error", *SIGGRAPH 1997 Course Notes*, August 1997.

45 D.P. Atherton, J.A. Bather and A.J. Briggs, "Data fusion for several Kalman filters tracking a single target," *IEE Proc.-Radar, Sonar Navigation*, vol. 152, no. 5, Oct. 2005, p. 372-376.

46 Y. He and E.K.P. Chong, "Sensor Scheduling for Target Tracking in Sensor Networks," *43rd IEEE Conference on Decision and Control*, 14-17 Dec 2004, p. 743-748.

47 Y. He and E.K.P. Chong, "Sensor Scheduling for Target Tracking: A Monte Carlo Sampling Approach," *Digital Signal Processing*, vol. 16 (2006), p. 533-545.

48 V. Krishnamurthy and B. Wahlberg, "Finite Dimensional Algorithms for Optimal Scheduling of Hidden Markov Model Sensors," *Proc. 2001 IEEE International Conference on Acoustics, Speech and Signal Processing*, 7-11 May 2001, vol. 6, p. 3973-3976.

49 Y. Li, L.W. Krakow, E.K.P. Chong and K.N. Groom, "Dynamic Sensor Management for Multisensor Multitarget Tracking," *Information Sciences and Systems, 2006 40th Annual Conference on*, March 2006, p. 1397-1402.

50 M. Rezacian, "Sensor Scheduling for Optimal Observability Using Estimation Entropy," <http://arxiv.org/cs/0609157> (2006).

51 V. Gupta, T. Chung, B. Hassibi and R.M. Murray, "Sensor Scheduling Algorithms Requiring Limited Computation," *Proc. of the 2004 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, 17-21 May 2004, p. 825-828 (2004).

52 T. Maddula, A.A. Minai and M.M. Polycarpou, "Multi-Target Assignment and Path Planning for Groups of UAVs," in *Cooperative Control and Optimization*, S. Butenko, R. Murphrey and P. Pardalos (eds.), Kluwer Academic Publishers, p. 261-272 (2004).

53 Y. Jin, M.M. Polycarpou and A.A. Minai, "Cooperative Real-Time Task Allocation Among Groups of UAVs," in *Cooperative Control and Optimization*, S. Butenko, R. Murphrey and P. Pardalos (eds.), Kluwer Academic Publishers, p. 207-224 (2004).

54 J.S. Baras and A. Bensoussan, "Optimal Sensor Scheduling in Nonlinear Filtering of Diffusion Processes," *SIAM J. Control and Optimization*, vol. 27, no. 4, p. 786-813, July 1989.

55 S. Ren, Q. Li, H. Wang, X. Chen and X. Zhang, "Design and Analysis of Sensing Scheduling Algorithms under Partial Coverage for Object Detection in Sensor Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 3, March 2007, p. 334-350.

56 J. Jeong, S. Sharafkandi and D. Du, "Energy-Aware Scheduling with Quality of Surveillance Guarantee in Wireless Sensor Networks," *International Conference on Mobile Computing and Networking, Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, p. 55-64, ACM Press, 2006.

57 M. Kalandros, L.Y. Pao and Y.-C. Ho, "Randomization and Super-Heuristics in Choosing Sensor Sets for Target Tracking Applications," *Proc. of the IEEE Conference on Decision and Control*, 7-10 Dec. 1999, vol. 2, p. 1803-1808.

58 S. Agnihotri and P. Nuggehalli, "Coding, Scheduling, and Cooperation in Wireless Sensor Networks," <http://arxiv.org/cs/0701051> (2007).

59 W.-D. Xiao, J.-K. Wu, L.-H. Xie and L. Dong, "Sensor Scheduling for Target Tracking in Networks of Active Sensors," *Acta Automatica Sinica*, vol. 32, no. 6, Nov. 2006.

60 T.J. Hestilow, T. Wei and Y. Huang, "Sensor Scheduling and Target Tracking Using Expectation Propagation," *Statistical Signal Processing, 2005 IEEE/SP 13th Workshop on*, 17-20 July 2005, p. 1232-1237.

61 W. Xiao, J.K. Wu and L. Xie, "Adaptive sensor scheduling for target tracking in wireless sensor network," *Proc. of SPIE*, vol. 5910, 2005.

62 A. Gage and R.R. Murphy, "Sensor Scheduling in Mobile Robots Using Incomplete Information via Min-Conflict with Happiness," *IEEE Trans. Systems, Man, and Cybernetics---Part B: Cybernetics*, vol. 34, no. 1, Feb 2004, p. 454-467.

63 G.A. McIntyre and K.J. Hintz, "Information theoretic approach to sensor scheduling," *Proceedings of SPIE -- Volume 2755, Signal Processing, Sensor Fusion, and Target Recognition V*, Ivan Kadar, Vibeke Libby, Editors, June 1996, pp. 304-312.

64 S. Maheswararajah and S. Halgamuge, "Sensor Scheduling for Target Tracking Using Particle Swarm Optimization," *Vehicular Technology Conference, IEEE 2006*, vol. 2, p. 573-577.

65 E.B. Ermis and V. Saligrama, "Distributed Detection in Sensor Networks with Limited Range Sensors," <http://arxiv.org/cs/0701178> (2007).

66 A.I. Mourikis and S.I. Roumeliotis, "Optimal Sensor Scheduling for Resource-Constrained Localization of Mobile Robot Formations," *IEEE Trans. Robotics*, vol. 22, no. 5, Oct. 2006, p. 917-931.

67 A. Talukder, S.-M. Ali, A. Panangadan, C. Jadhav, R. Pidva, R.K. Bhatt, L. Chandramouli and S. Monacos, "Optimal Sensor Scheduling and Power Management in Sensor Networks," *Optical Pattern Recognition XVI*, edited by D.P. Casasent, T.-H. Chao, *Proc. of SPIE*, Vol. 5816 (SPIE, Bellingham, WA, 2005), p. 221-232.

68 S. Martinez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, (2006), p. 661-668.

69 V. de Silva and R. Ghrist, "Coverage in Sensor Networks via Persistent Homology," *Algebraic & Geometric Topology*, vol. 7, 2007, p. 339-358

70 V. de Silva and R. Ghrist, "Coordinate-free Coverage in Sensor Networks with Controlled Boundaries via Homology," *International J. of Robotics Research*, vol. 25, no. 12, Dec. 2006, p. 1205-1222.

71 B. Srivastava and E. Klassen, "Bayesian, Geometric Filtering for Subspace Tracking," *Applied Probability Trust* (27 August 2001), <http://lcv.stat.fsu.edu/publications/paperfiles/ptpaper.pdf> (2001).

72 J. Cortes, S. Martinez and F. Bullo, "Spatially-Distributed Coverage Optimization and Control with Limited-Range Interactions," <http://arxiv.org/math/0401297> (2004).

73 D. Smith and S. Singh, "Approaches to Multisensor Data Fusion in Target Tracking: A Survey," *IEEE Trans. Knowledge and Data Engineering*, vol. 18, no. 12, Dec. 2006, p. 1696-1710.

74 D.R. Keller, T.K. Moon and J.H. Gunther, "Source Localization from Moving Arrays of Sensors," *Signals, Systems and Computers, 2005, 39th Asilomar Conference on*, 28 Oct – 01 Nov 2005, p. 452-456.

75 N. Okello and D. Musicki, "Emitter geolocation with two UAVs", in Proc. 2007 Information, Decision and Control Conference, (Adelaide, Australia), February 2007.

76 K. C. Ho and Y. C. Chan, "Solution and performance analysis of geolocation by TDOA", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No.4, October 1993, pp.1311-1322.

77 N. Okello, "Geolocation with multiple UAVs", in Proc. 9th International Conference on Information Fusion, (Florence, Italy), July 2006.

78 D. Musicki and R. Evans, "Measurement Gaussian sum mixture target tracking", in Proc. 9th International Conference on Information Fusion, (Florence, Italy), July 2006.

79 D. Musicki, R. Evans, and B. La Scala, "Integrated track splitting suite of target tracking filters", Proc. 6th International Conference on Information Fusion, (Cairns, Queensland, Australia), July 2003.

80 D. J. Salmond, "Mixtures reduction algorithms for target tracking in clutter", *Signal and Data Processing of Small Targets 1990*, Proc. SPIE, Vol. 1305}, April 1990.

81 S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.